物理ベースマテリアルのためのマイクロファセットBRDF入門 ～リアルな光表現の本質～

2023年8月25日

石山 浩平

大垣 真二

それでは、物理ベースマテリアルのためのマイクロファセットBRDF入門～リアルな光表現の本質～というタイトルで講演を行います。

講演者の紹介

石山浩平
株式会社ポリフォニー・デジタル
エンジニア

大垣真ニ
所属組織における業務ではなく、今回は個人での活動です

CEDEC2024

本公演は、株式会社ポリフォニー・デジタルのエンジニア、石山。そして大垣の二名が行います。

講演の内容

- 前半
  - 光反射の物理
  - マイクロファセットBRDF

- 後半
  - 法線分布関数の詳細
  - 最新の研究紹介

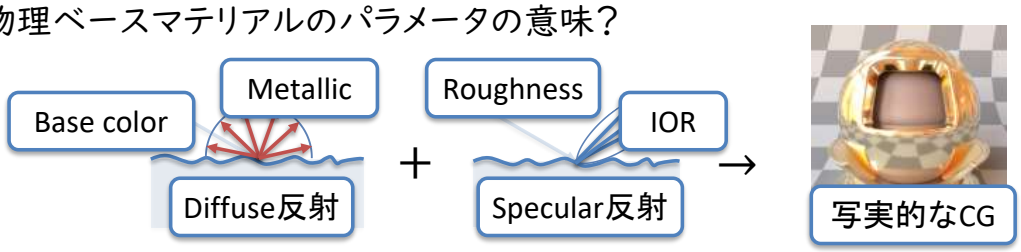
講演は前半と後半に分けて行います。

前半では、物理ベースレンダリングを背景として、「光反射の物理」「マイクロファセットBRDF」の基礎的な内容をお話しします。

後半では、やや難しめの内容となりますが、マイクロファセットBRDFを構成する要素である、法線分布関数の詳細と、最新の研究紹介を行います。

はじめに

- 物理ベースマテリアルのパラメータの意味？



- Specular反射を表す、マイクロファセットBRDFとは？

$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$

それでは、前半パートを始めます。

ゲームエンジンやレンダラーで物理ベースマテリアルをいじると、何やらDiffuse反射とSpecular反射というものがあり、それを元に写実的な絵が出ることが、なんとなくわかります。これらの反射率は「Base color」「Metallic」「Roughness」「IOR」といったパラメータで調整することができます。まずは、それぞれのパラメータの物理的な意味を、物理の基礎から出発して考えます。

その後、物理ベースレンダリングを実装や改造しようと考えたときに立ちはだかる、マイクロファセットBRDFの構成を紹介します。これにより、ラフネスがどういう概念なのかを、詳しく知ることができます。

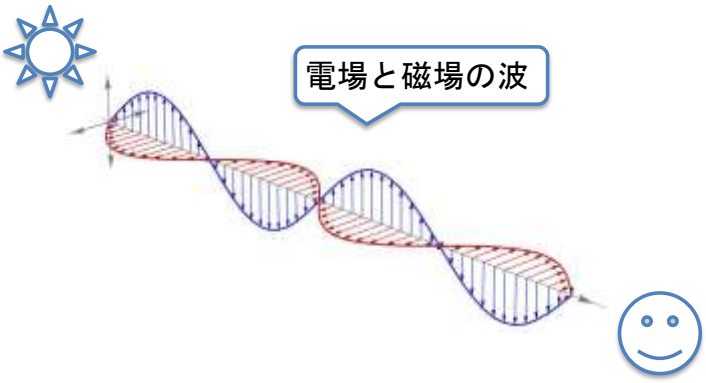
光と物質

それでは物理の基礎に立ち戻り、光とは何か、物質とはなにかをおさらいします。

参考文献

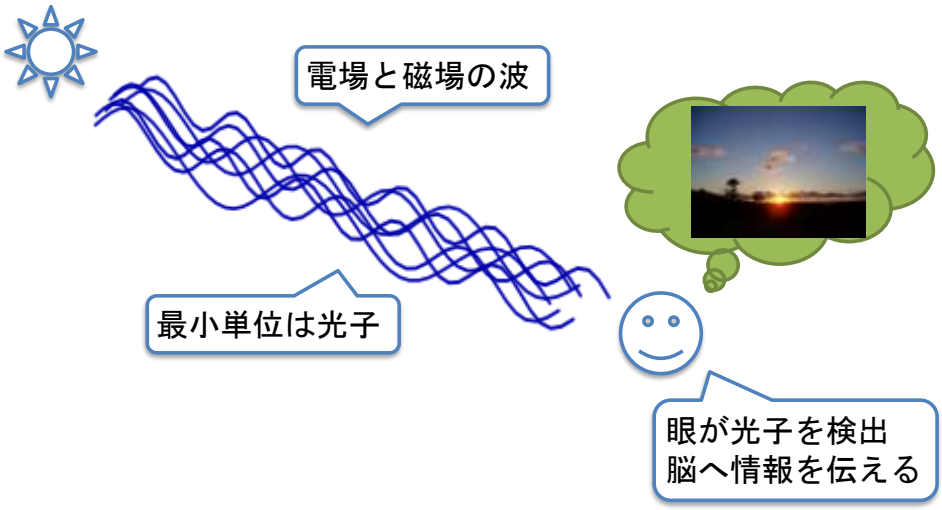
- J.J.サクライ著, 樺沢 宇紀訳. (2010) *サクライ「上級量子力学」第 I 巻.*
- 田所 利康, 石川 謙, 大津 元一. (2014). *イラストレイテッド光の科学.*
- Heckt, E. (2015) *Optics. 5th ed.*

光は電磁波



光とは、電磁波のことでした。電磁波とは、電場と磁場の波のことです。

光は電磁波



以後、電磁波をひとつの波線として簡略的に表記します。

(click)

自然にある光源は、様々な波長の電磁波が放射します。

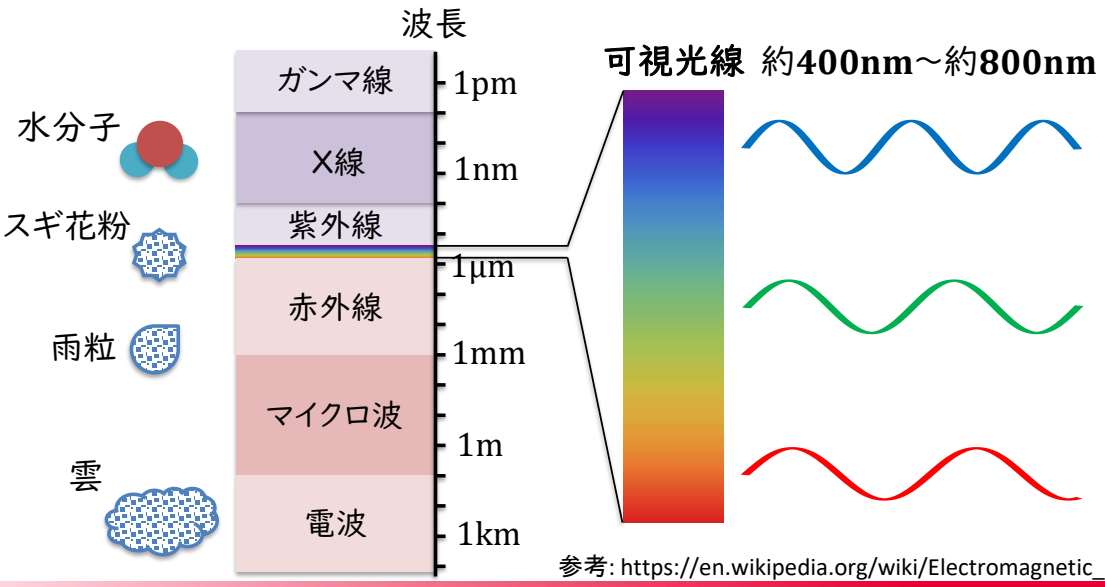
(click)(click)..

その最小単位は、光子と呼ばれます。

(click)

我々の眼は、光子を検出して、その情報を電気信号として脳に伝えることで、目の前の風景を認識しています。我々のスケールでは光子はとてもたくさんあるものなので、以後は、その集団的な現象である電磁波について考えます。それでは、我々の眼が検出する電磁波はどういうものなのか？

可視光線

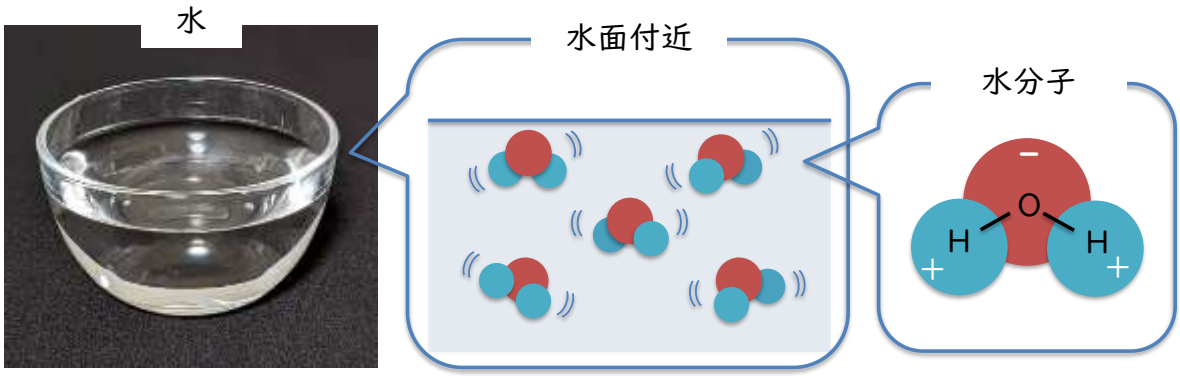


それは、400nm～800nmの波長をもつ電磁波です。これは可視光線と呼ばれています。

(click)

この波長がどれくらいのものかというと、雨粒よりずいぶん小さく、水分子よりずいぶん大きい、スギ花粉の粒子と同じくらいの長さです。次に、その雨粒やスギ花粉、水分子とは何か、つまり物質とは何だったかをおさらいします。

物質



水を例にとります。

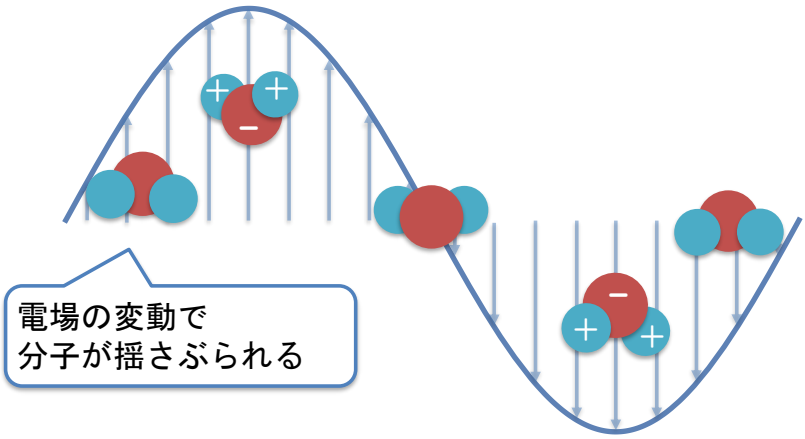
(click)

物質とは、原子や分子が集まってできているものでした。

(click)

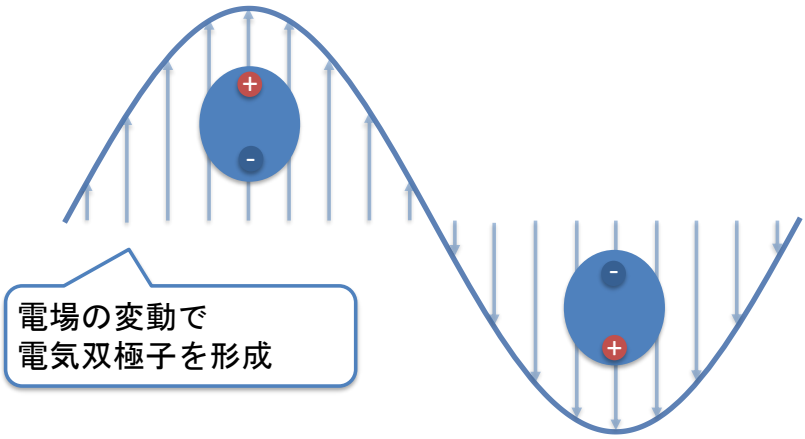
そして、分子は原子が集まってできたものです。たとえば、水分子は一つの酸素原子と、二つの水素原子によって成り立っています。原子や分子は、電荷がマイナスの電子をもちます。つまり、その内部に電荷分布をとる性質があります。次に、物質に光が当たったとき、つまり原子や分子が電磁波にさらされたら何が起こるかを考えます。

電気双極子



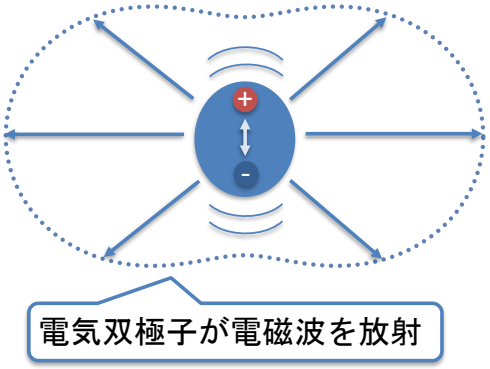
電磁波は電場と磁場の波でしたから、それにさらされた原子や分子は揺さぶられます。つまり、電子が動いて電荷分布を変えます。

電気双極子



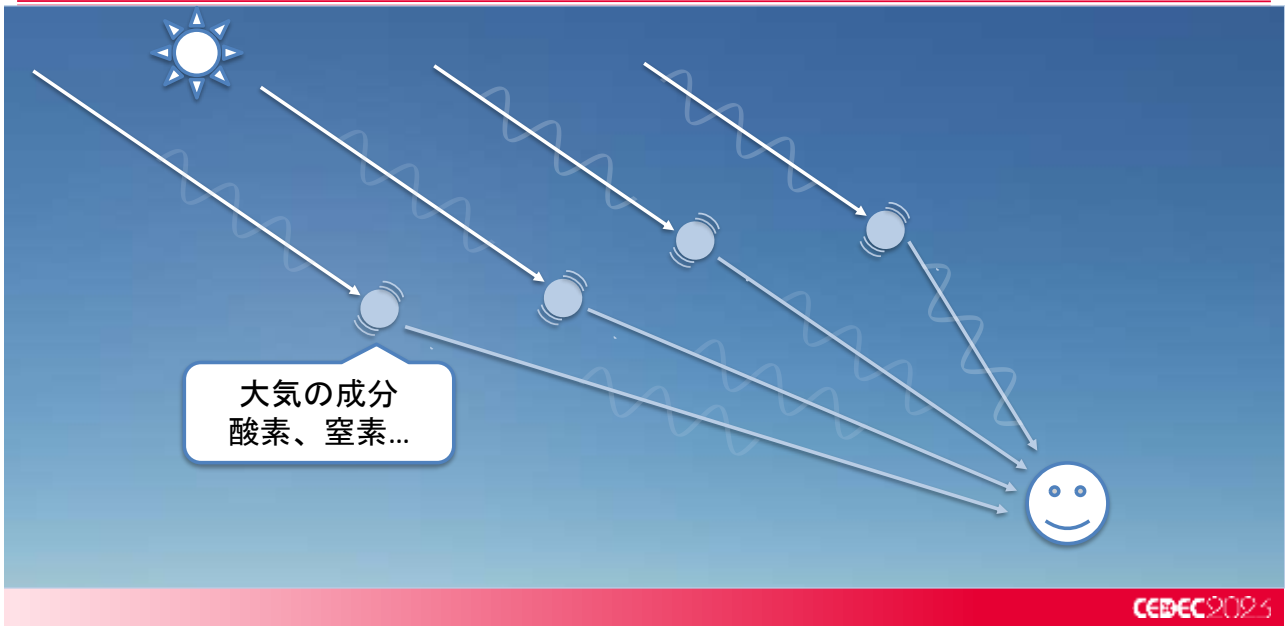
このことを抽象的にとらえると、電磁波にさらされた原子や分子は電気双極子を形成します。

レイリー散乱



そうして形成された電気双極子は、電磁波を放射します。このように、原子や分子など小さな粒子に光が当たって電気双極子を形成し、電磁波を放射する現象はレイリー散乱と呼ばれます。我々は、レイリー散乱をよく見慣れています。

レイリー散乱と空の色



何故なら、空で起きているからです。空、つまり大気は酸素分子と窒素分子を主成分としてできており、それに太陽光が当たることでレイリー散乱を起こします。レイリー散乱は光の波長が短いほど散乱が起きやすい、という性質があるので、われわれの眼は可視光線のうち波長の短い色、つまり青色を空の色として認識しているわけです。

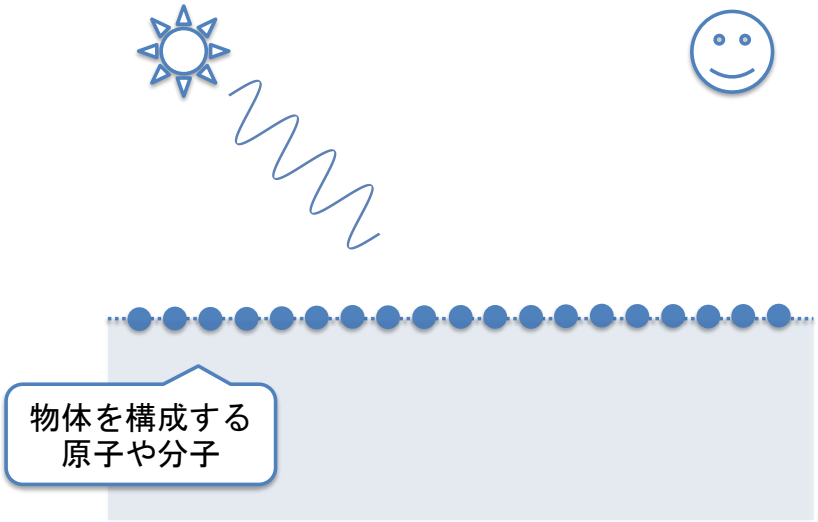
反射と屈折

つぎに、物体の表面で起こる反射や屈折を考えます。

参考文献

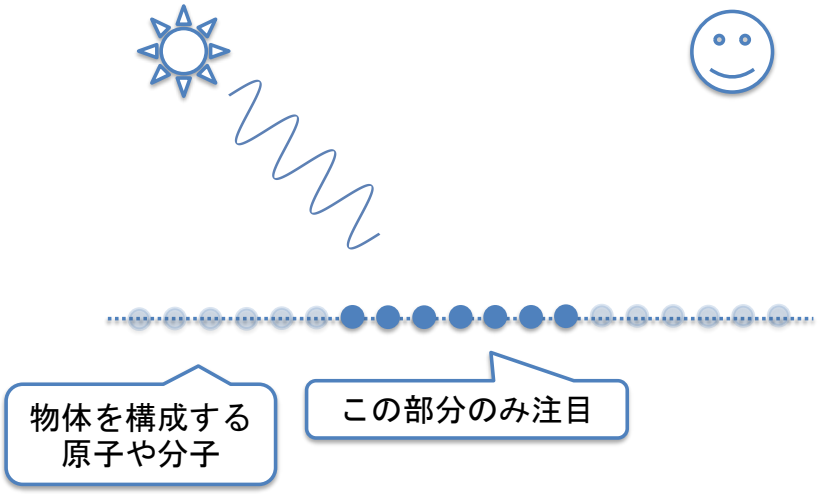
- Zangwill, A. (2013) *Modern electrodynamics.*
- 田所 利康, 石川 謙, 大津 元一. (2014). *イラストレイテッド光の科学.*
- Heckt, E. (2015) *Optics. 5th ed.*
- Hoffman, N. (2020). *Some Thoughts on the Fresnel Term.*

電磁波の反射 / ホイヘンスの原理



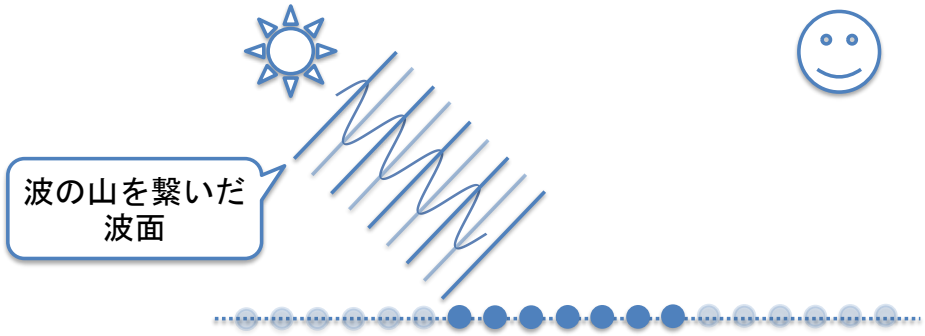
高校物理で学ぶ、ホイヘンスの原理に基づいて説明しましょう。図のように、物体を構成する原子や分子を簡略的に表します。ここへ入射する電磁波は、表面へ満遍なく当たるものですが…

電磁波の反射 / ホイヘンスの原理



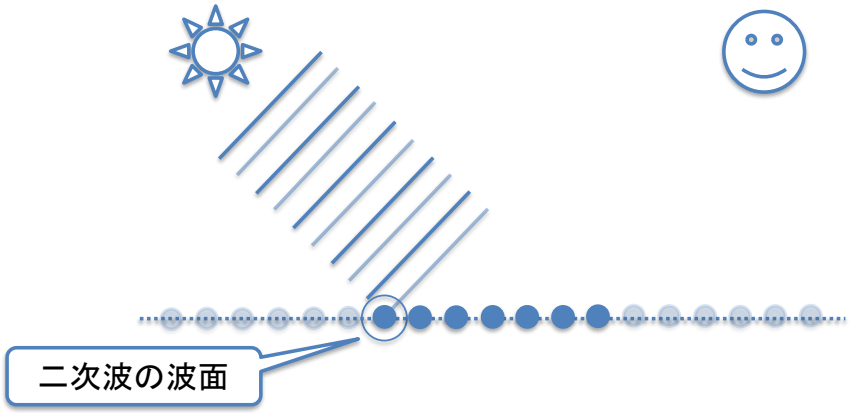
簡単のために、この部分のみ注目します。

電磁波の反射 / ホイヘンスの原理



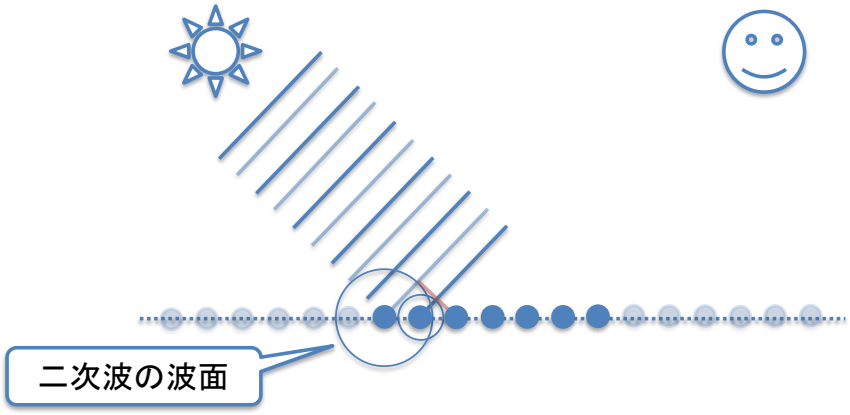
光源が放射する電磁波の波面を、
(click)(click)
このように線として表します。そして電磁波が表面へ進んでいき..
(click)(click)..

電磁波の反射 / ホイヘンスの原理



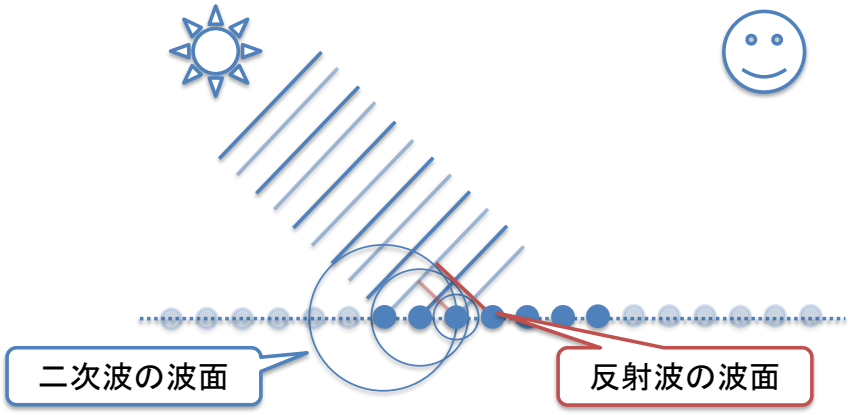
表面に到達すると、その表面の物質が光を散乱して二次波を形成します。

電磁波の反射 / ホイヘンスの原理



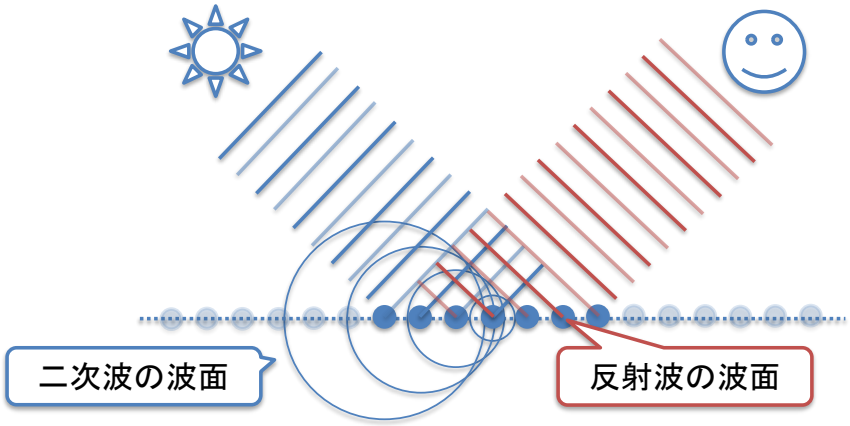
二次波は等方に放射されるものとして、波面を球で表します。

電磁波の反射 / ホイヘンスの原理



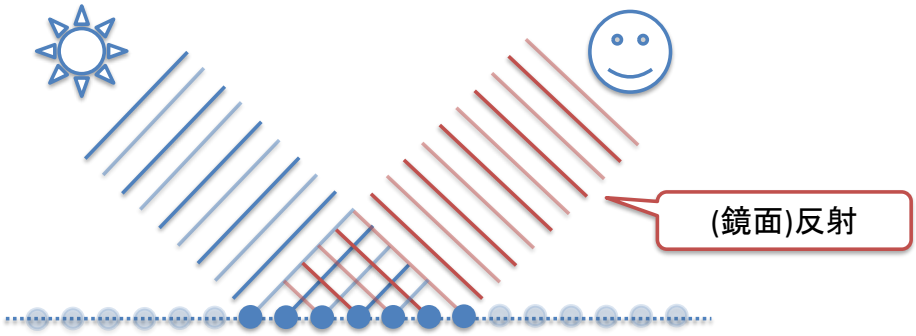
二次波の波面をつなぐと、反射波が現れることがわかります。

電磁波の反射 / ホイヘンスの原理



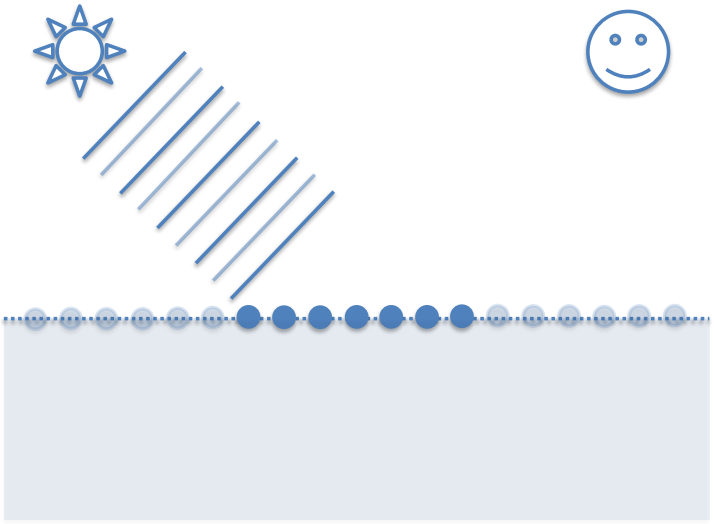
(click)(click)..

電磁波の反射 / ホイヘンスの原理



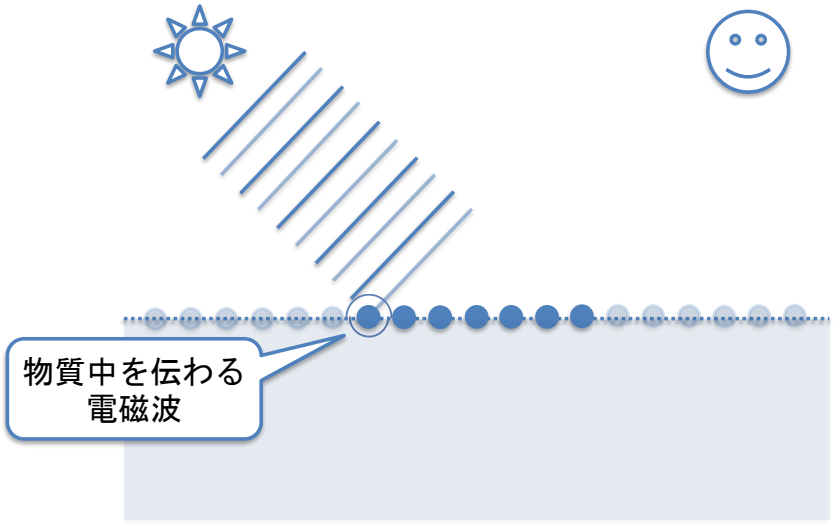
こうして、表面上で起きる2次波を重ね合わせた結果として、鏡面反射が起きることがわかりました。

電磁波の屈折 / ホイヘンスの原理



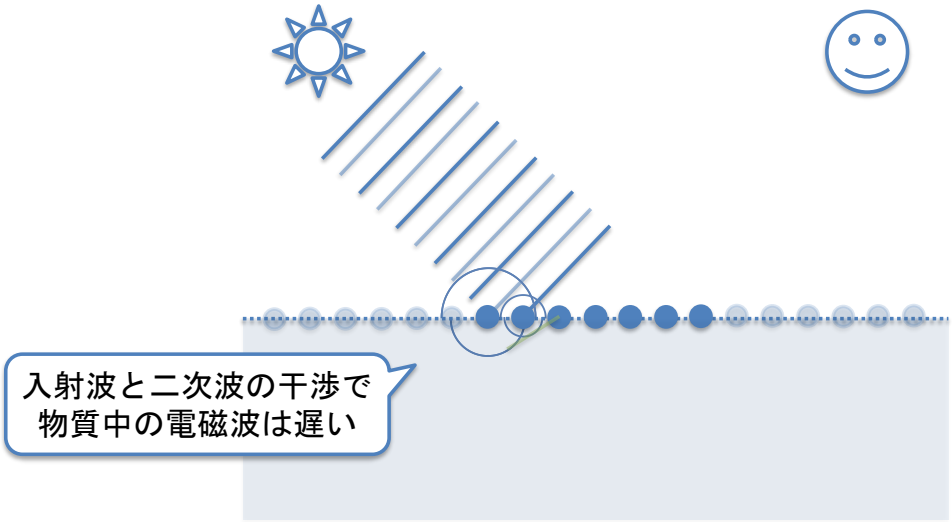
次は、屈折を考えます
(click)(click)..

電磁波の屈折 / ホイヘンスの原理



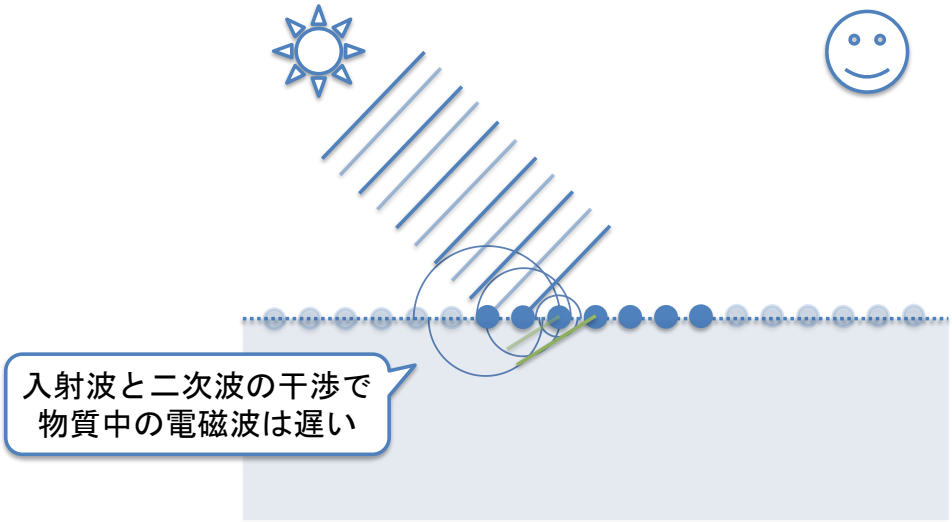
いま、物質中に伝わる電磁波に注目します

電磁波の屈折 / ホイヘンスの原理



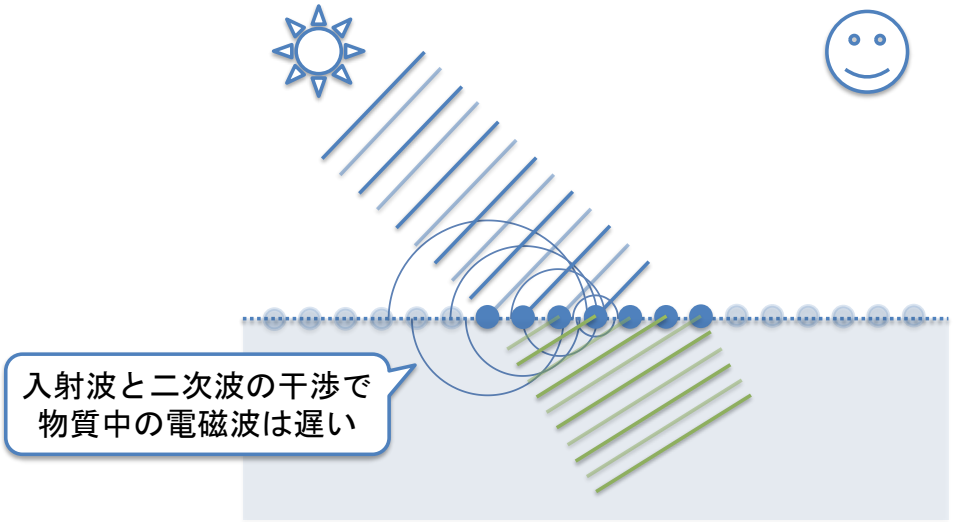
物質中の電磁波は、入射波と二次波の干渉などの効果により、その伝搬速度が遅くなります。

電磁波の屈折 / ホイヘンスの原理



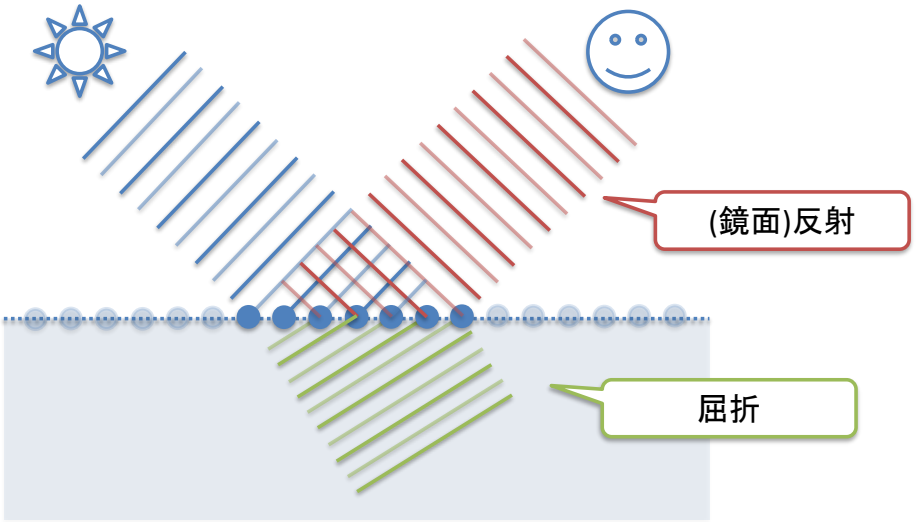
この効果を表現するため、図の下側の円は、上側に比べて小さく描きます

電磁波の屈折 / ホイヘンスの原理



さきほどと同様にして波面を結ぶと、物質内を屈折して伝わる電磁波が起こることがわかります。

電磁波の反射と屈折 / ホイヘンスの原理



反射波と、屈折波を合わせて書くとこんな感じです。このようにして、物体の表面に電磁波が入射すると、鏡面反射と屈折がおきることがわかりました。

この、屈折の仕方の違いは、物質ごとに異なります。その違いを表す物理量が…

屈折率 Index of Refraction (IOR)

屈折率です。英語では、Index of Refraction、略して IOR と呼ばれます。

この量は、物質中の光速、つまり電磁波の速度と、真空中の光速の比で定義されます。

(click)

空気中でだいたい 1。

(click)

水がだいたい 1.33。

(click)

ガラスがだいたい 1.5。

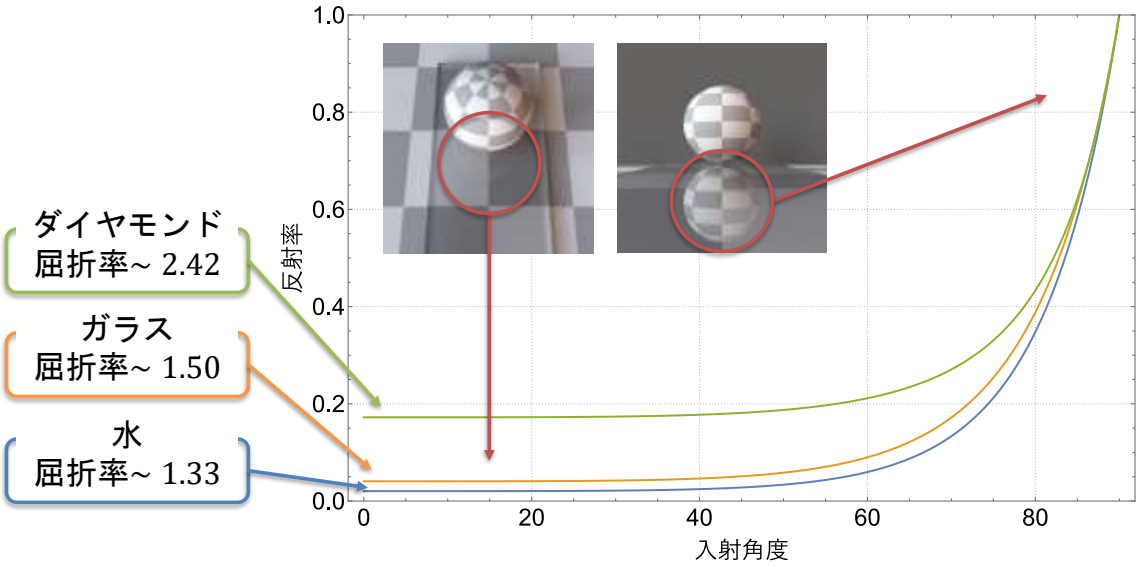
(click)

ダイヤモンドがだいたい 2.4 です。

屈折率の違いによって、それぞれの球に映る像のゆがみ方が変わることがわかります。ここで、ダイヤモンドの球に注目すると、ほかに比べて光の反射率が高い様子が見て取れます。

反射率や屈折率は、屈折率の異なる境界で起きる電磁波の反射と屈折を考察して得られる、フレネルの式で得られることがわかっています。

フレネル反射率(偏光なし)



その様子がこちらです。これによると、ガラスや水の反射率はだいたい 4% くらい。ダイヤモンドはそれよりも高くなることがわかります。

このほか、ガラスを眺める角度で反射率がかわることもわかります。グラフの左側にあるように、ガラスを上から眺めたときは反射率が低いですが、眺める角度が水平に近くなるにしたがって反射率が高くなることがわかります。

こうして、屈折率のことなる様々な物質をCGで表現できるようになりました。

ここにうさぎさんを置いておきます。このCGは Mistuba 3 というレンダラーを使って手元で生成したものです。

つぎは、金属のうさぎさんを作ることを考えます。このように、透明な物体と、金属光沢をもつ物体の性質の違いはどこから来るのでしょうか。

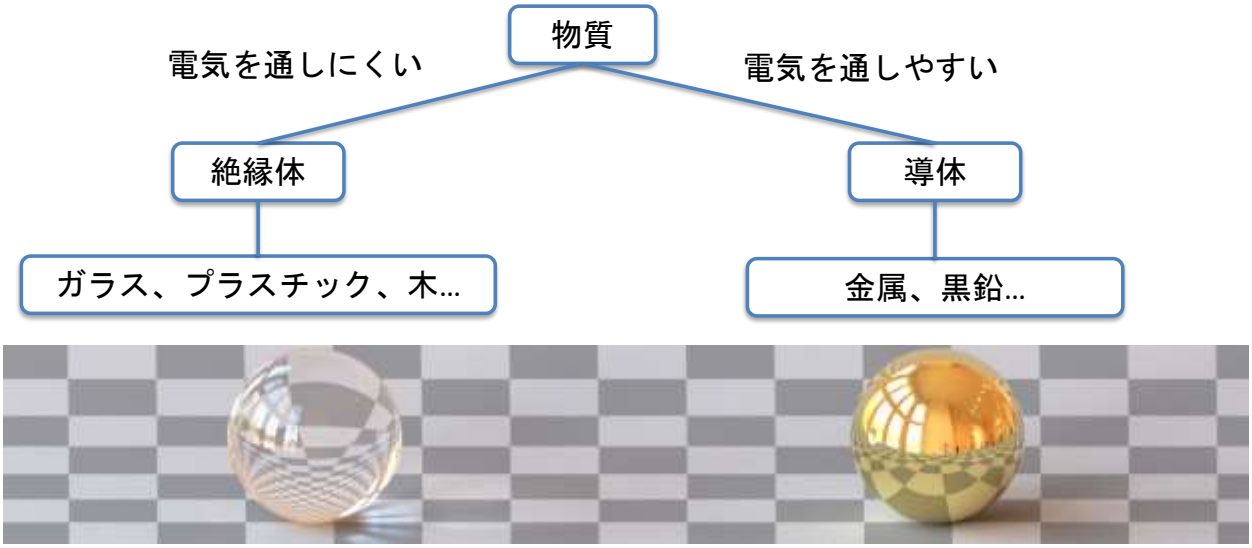
絶縁体と導体

それは、絶縁体と導体の違いです。

参考文献

- Zangwill, A. (2013) *Modern electrodynamics.*
- 田所 利康, 石川 謙, 大津 元一. (2014). *イラストレイテッド光の科学.*
- Heckt, E. (2015) *Optics. 5th ed.*
- Babar, S. and Weaver, J. H. (2015). *Optical constants of Cu, Ag, and Au revisited.*
- Hoffman, N. (2020). *Some Thoughts on the Fresnel Term.*

絶縁体と導体



我々の世界にある物質を、電気を通すか通さないかで二つに分けます。このように分ける理由は、両者で光反射の特性が大きく異なるからです。

左側の、電気を通しにくい物質は絶縁体と呼ばれます。ガラスやプラスチック、木などです。

右側の、電気を通しやすい物質は導体と呼ばれます。ふだん我々が見慣れているものでは、金属があります。

両者の電気の通しやすさの違いがどこから来るかというと…

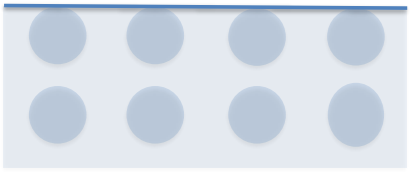
※補足

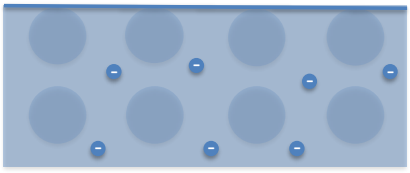
電気伝導率で物質を分類したとき、その文脈に応じて超伝導体、導体(導電体、電気伝導体)、半導体、誘電体、絶縁体に分けられることが一般的です。この資料では、ざっくり導体・絶縁体の二種類に分けて説明を行っています。このため、超伝導体や半導体の分類は行わず、また絶縁体と誘電体を一緒にして扱っています。

絶縁体と導体

絶縁体
自由電子がほとんどない

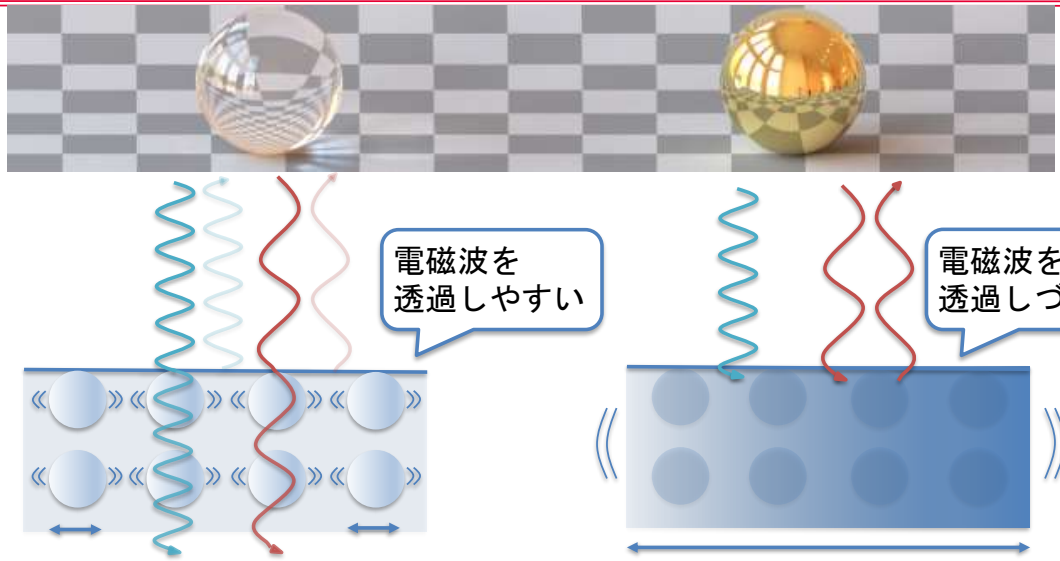
導体
自由電子がたくさんある





自由電子がたくさんあるか、ないかによって決まります。絶縁体は自由電子がほとんどないので、電気もほとんど通らないですが、導体は自由電子がたくさんあるので、電気を通しやすいです。この違いにより、光の反射、つまり入射した電磁波に対する応答に違いが出ます。

絶縁体と導体

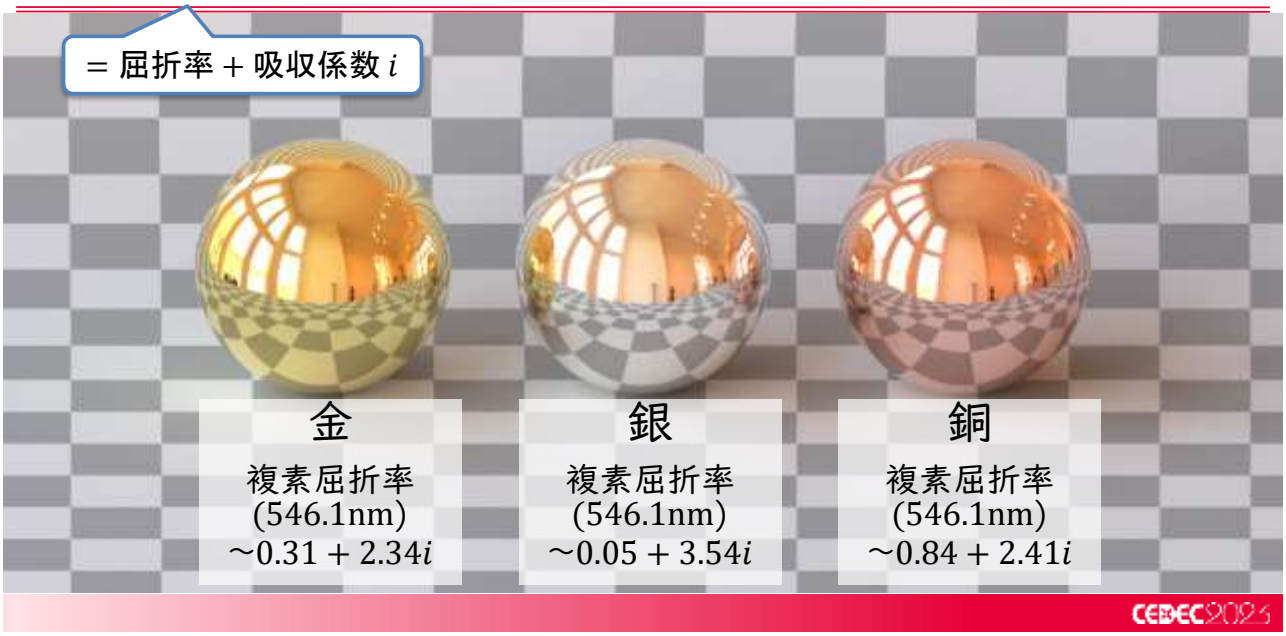


右側の絶縁体は、先ほど見たように電磁波を透過・屈折します。左側の導体は、電磁波が内部に入りづらいです。

その物質に対して短めの電磁波のエネルギーは、自由電子が吸収して熱に変えます。長めの電磁波のエネルギーは、自由電子が吸収してほとんどが表面付近で反射します。

このように、導体の光反射の性質を記述する物理量は…

複素屈折率



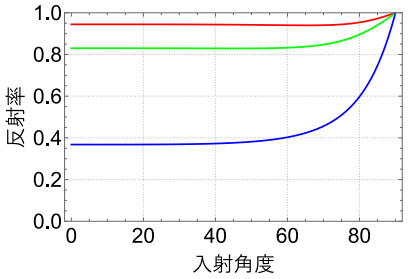
複素屈折率です。複素屈折率は、屈折率に吸収率を合わせた量で、複素数で書かれます。

(click)

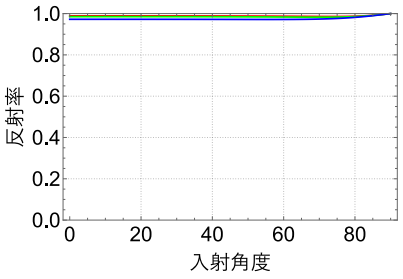
ここに、金・銀・銅の値を参考に書いておきます。それぞれ、緑色あたりの波長の複素屈折率の値です。複素屈折率からも、反射率を計算することができます。

フレネル反射率

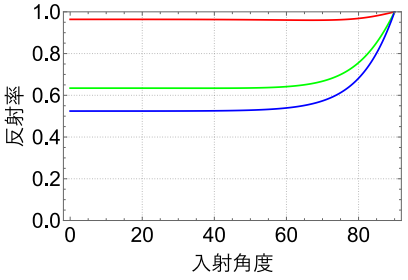
金



銀



銅



それがこのグラフです。金・銀・銅それぞれのRGBごとの反射率を書きました。これを見ると、ガラスや水といった絶縁体の 4%程度の反射率に比べれば、ずいぶん反射率が高いことがわかります。

こうして、反射率と複素屈折率を使って、透明な絶縁体と、金属を表現できるようになりました。

次は、一番下のウサギさんを考えます。この見た目は、絶縁体の中に光が入り、多重散乱して出てくることによってつくられます。

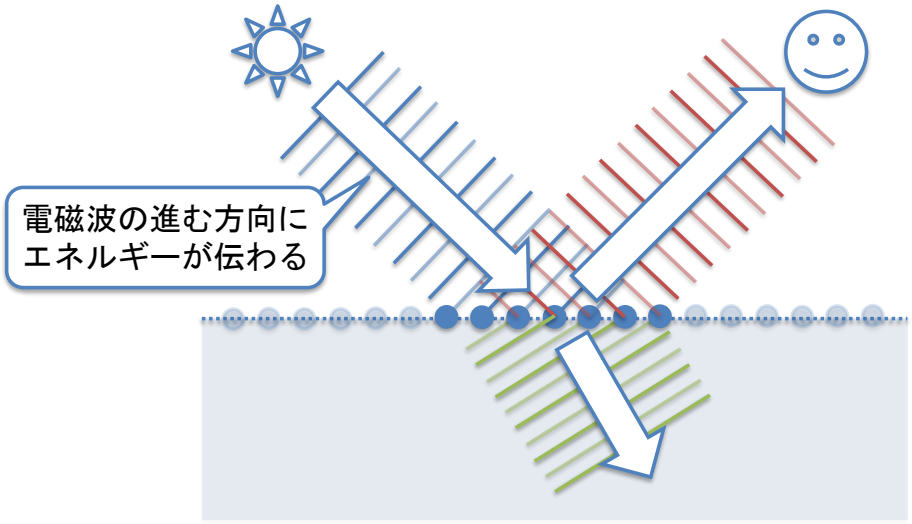
光線

多重散乱を考える前に、「光線」という概念を導入しましょう。

参考文献

- Born, M. and Wolf, E. (1999). *Principles of Optics, 7th ed.*

電磁波と光線



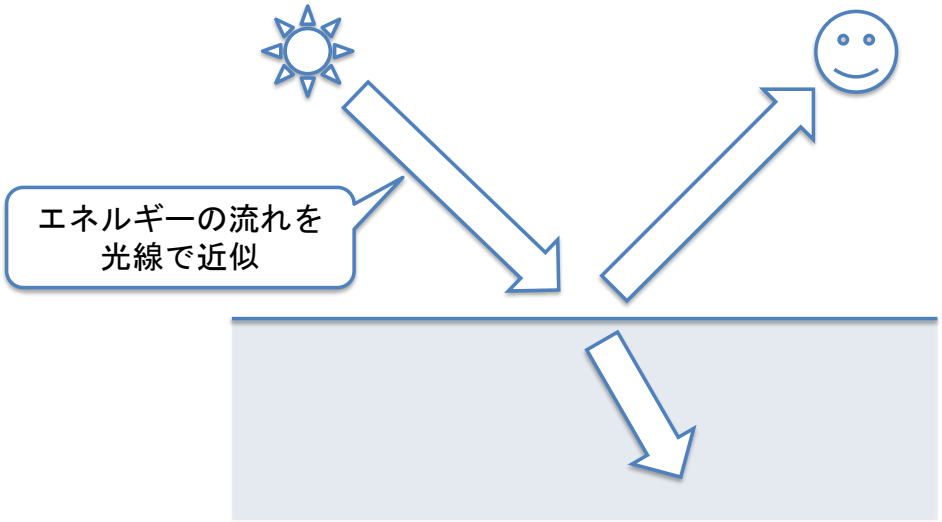
電磁波とは、電場と磁場の波でした。そして、電磁波の伝わる方向にエネルギーが伝わります。エネルギーとは、電荷を動かす量のことです。

電磁波の伝わる方向に線をひくと…

(click)

光線があらわれます。

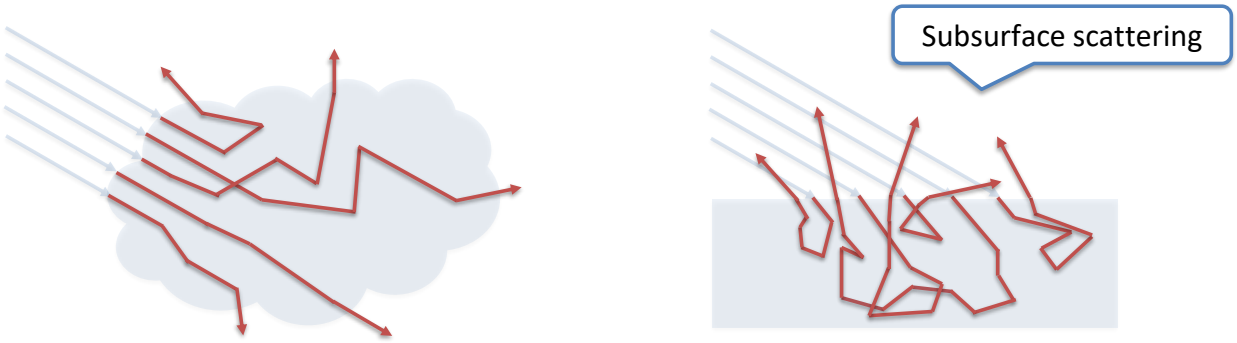
光線



そして、エネルギーの流れを光線で近似します。近似するとは、干渉など電磁波の波動的な性質を無視するということです。

この光線の概念を使って、絶縁体の中で起こる多重散乱を考えましょう。

絶縁体の内部散乱



まず、左側の雲のようなうさぎさんを考えます。

(click)

この雲の中に光線が入射すると、

(click)(click)..

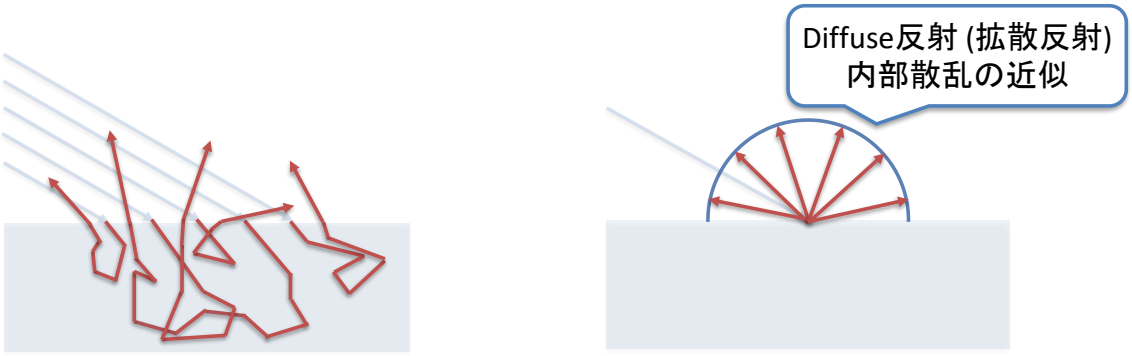
さまざまな方向に散乱して出ていきます。

このうさぎさんを右側のようにギュッと固めると、先ほどの現象が表面付近で起こります。この現象は、subsurface scattering と呼ばれます。

参考文献：Radiative Transfer方程式の拡散近似

- Jensen, H. W., et al. (2001). *A practical model for subsurface light transport*.

Diffuse反射：内部散乱の近似



subsurface scattering をさらに近似することを考えます。

(click)

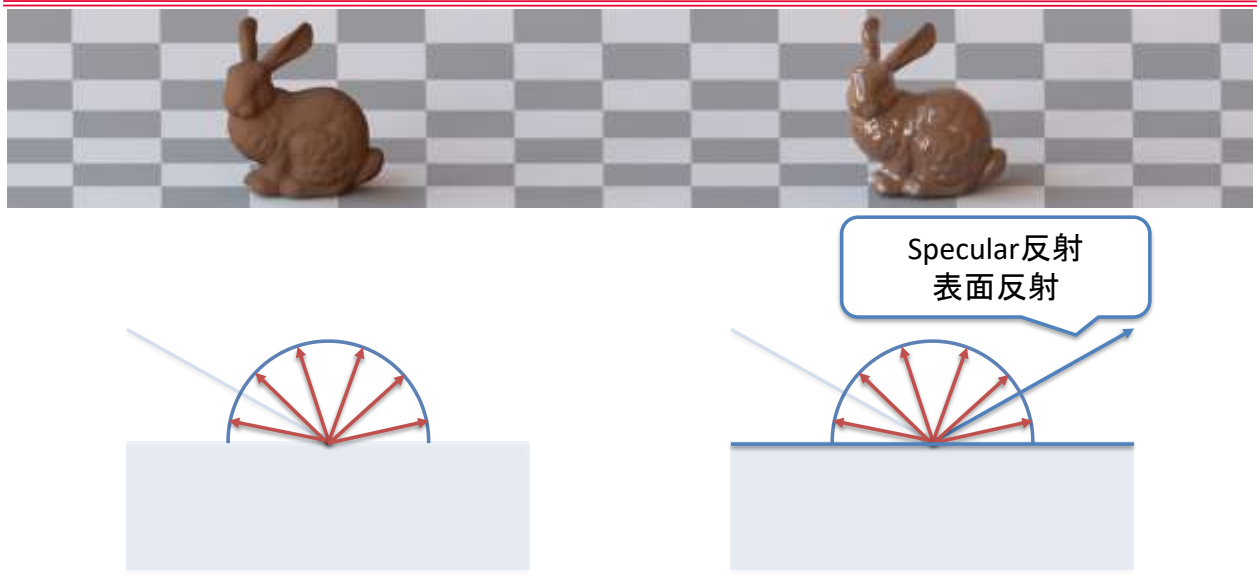
それには、拡散反射を考えます。こうすることで、たくさんの光線を計算しなくても、絶縁体の内部散乱を近似することができます。

この反射を Diffuse 反射、あるいは拡散反射と呼びます。

参考文献

- Akenine-Moller, T., et al. (2018). *Real-time rendering 4th ed.*
- Pharr, M., et al. (2018). *Physically based rendering: From theory to implementation 4th ed.*

Specular反射：表面反射



Diffuse反射だけだと、物体の内部散乱しか表現できないので、表面で起こる反射を加えます。

(click)

そして、表面で起きる反射を

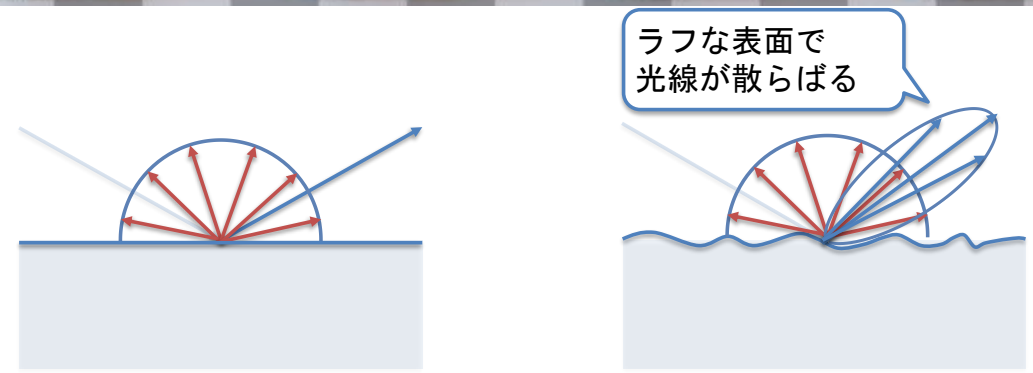
(click)

Specular反射と呼びます。このままではツルツルな表面しか表現できないので…

参考文献

- Akenine-Moller, T., et al. (2018). *Real-time rendering 4th ed.*
- Pharr, M., et al. (2018). *Physically based rendering: From theory to implementation 4th ed.*

Roughness：表面の粗さ



CEDEC2024

ラフネスというパラメータに応じて、表面がざらざらになり、

(click)

反射する光線の方向が散らばるようにします。

(click)

これで、ツルツルな表面から、ザラザラな表面までを表現できるようになりました。

参考文献

- Akenine-Moller, T., et al. (2018). *Real-time rendering 4th ed.*
- Pharr, M., et al. (2018). *Physically based rendering: From theory to implementation 4th ed.*

こうして、透明な絶縁体、そして金属に加え、不透明な絶縁体を表現できるようになりました。一番左のうさぎさんのように、ざらざらな物体も表現できるようにもなりました。

つまり、Diffuse反射とSpecular反射、そしてラフネスを考えることで、我々が目にする多くの物質、すなわち絶縁体と導体を近似的に表現できるということです。

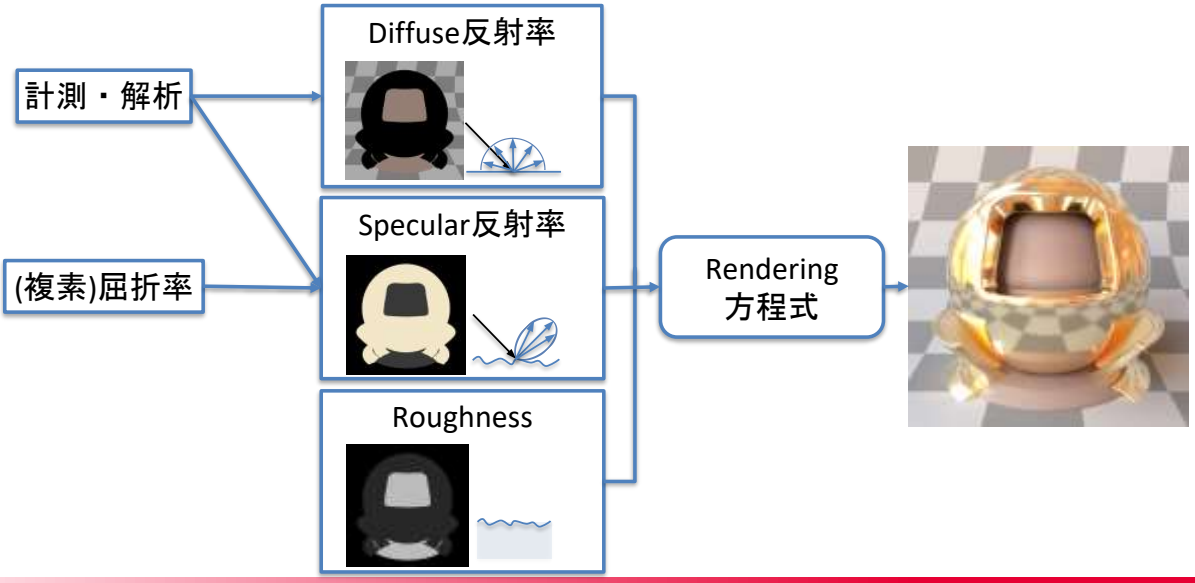
Diffuse/Specular
と
Base color/Metallic

以上のことを踏まえ、物理ベースマテリアルに使われるパラメータについて考えます。

参考文献

- Adobe Inc. (2023). *THE PBR GUIDE - PART 1.*
- Adobe Inc. (2023). *THE PBR GUIDE - PART 2.*

Diffuse / Specular



まず、Diffuse反射は内部散乱を表し、Diffuse反射率でその量を設定します。そして、Specular反射は表面反射を表し、Specular反射率でその量を設定します。

Roughness は、表面の粗さを変える量です。

(click)

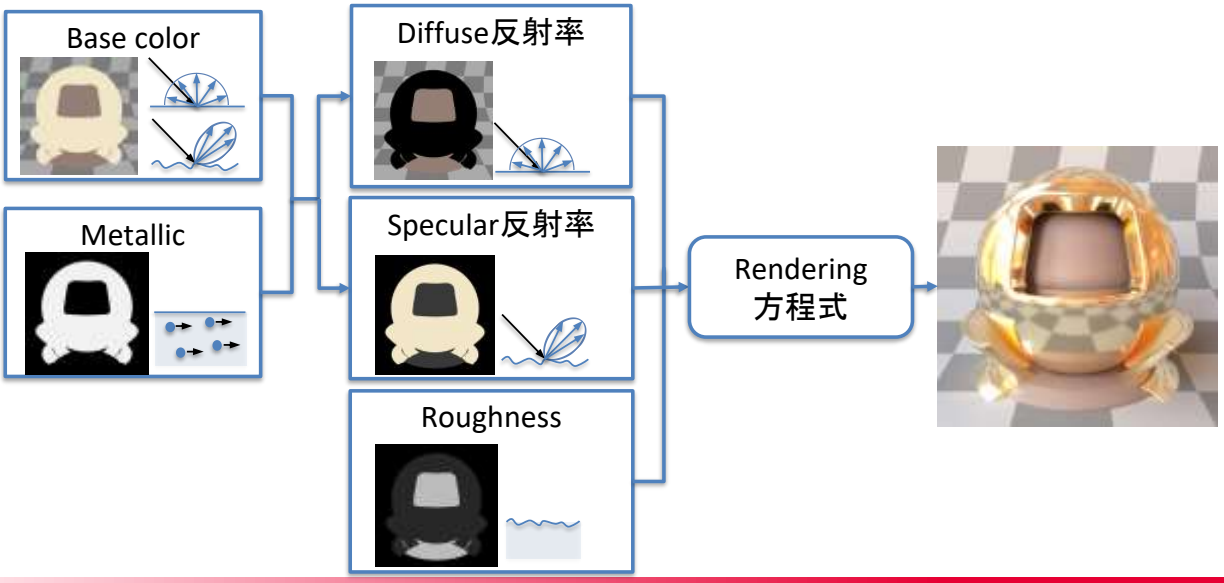
Specular反射率は、屈折率、あるいは複素屈折率から求めることができます。

(click)

Diffuse反射率は多重散乱ゆえに、単純に求めることは難しいですが、計測や解析、あるいは手で与えることで設定できます。

これらの反射率は、良く知られた別の表現があります。

Base color / Metallic



それは、Base color と Metallic です。

Base color には、Diffuse 反射率と Specular 反射率の両者が入っています。Metallic は、どこが金属なのか、つまり絶縁体か導体かを区別するパラメータです。この両者から、Diffuse反射率とSpecular反射率を換算することができます。

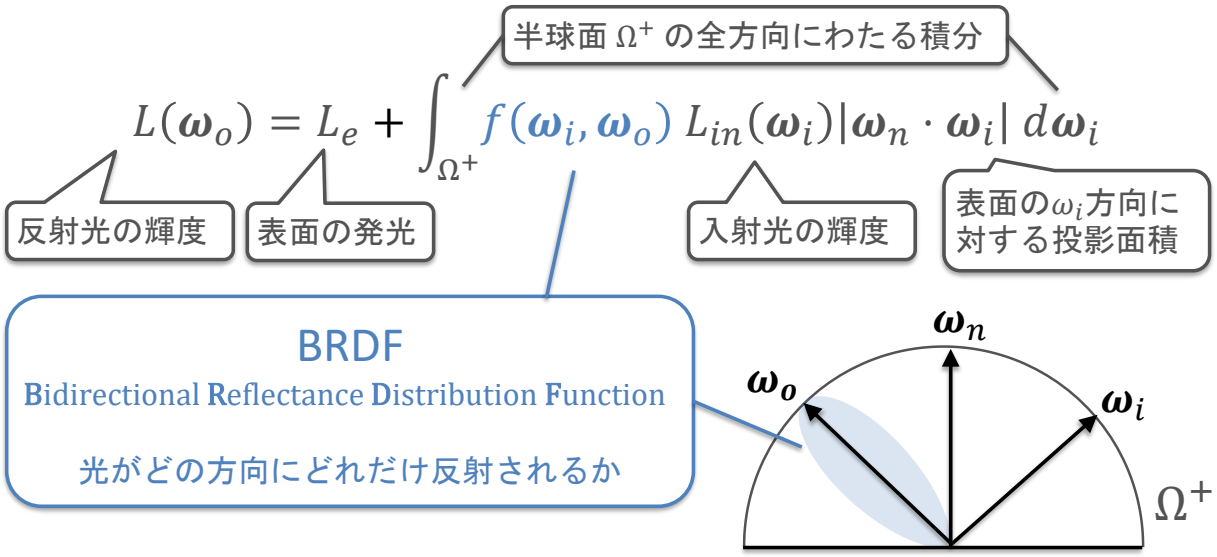
こうして得られた Diffuse反射率とSpecular反射率、ラフネスを Rendering 方程式で計算することで、写実的な映像が作り出されます。以降のスライドでは数式を使って、この計算を自分で作りたいとなったときに立ちはだかる、Rendering方程式を簡単に説明します。

Rendering方程式

参考文献

- Akenine-Moller, T., et al. (2018). *Real-time rendering 4th ed.*
- Pharr, M., et al. (2018). *Physically based rendering: From theory to implementation 4th ed.*

Rendering方程式



Rendering方程式は、このような形をしています。

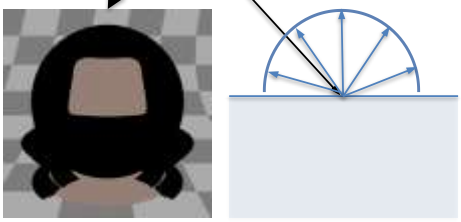
(click)

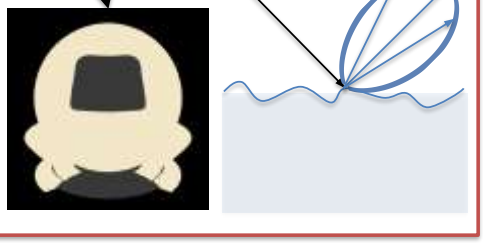
おおまかには、入射光にBRDFをかけて反射光を求める式です。色々な要素からできていますが、ここでは詳しい内容は割愛して、BRDFのみに注目します。

BRDFとは、表面へ入射した光が、どの方向にどれだけ反射されるかを表した分布関数です。それぞれの物質の反射特性は、このBRDFによって表現されます。

Diffuse反射とSpecular反射

$$f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \rho_{\mathrm{d}}\, f_{\mathrm{diffuse}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) + \rho_{\mathrm{s}}\, f_{\mathrm{specular}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$$





Diffuse反射とSpecular反射と合わせたBRDFはこのようなものです。

(click)

Diffuse反射率。

(click)

Diffuse反射の分布関数。

(click)

Specular反射率。

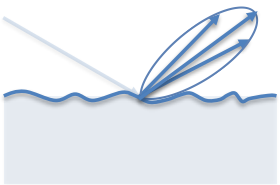
(click)

Specular反射の分布関数。

(click)

以後は、Specular反射に注目します。このスペキュラー反射を作る分布関数の一つとして…

マイクロファセットBRDF



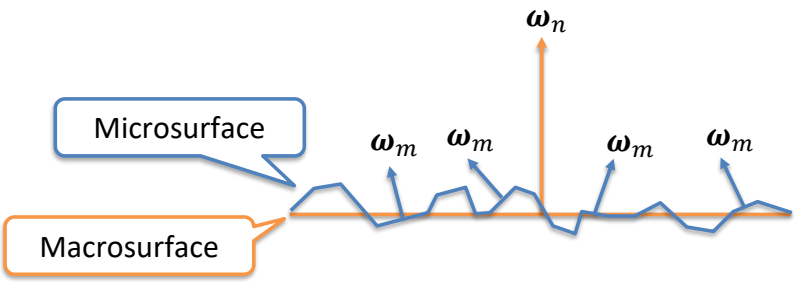
マイクロファセットBRDFが知られています。

参考文献

- Walter, B., et al. (2007). *Microfacet models for refraction through rough surfaces.*
- Heitz, E. (2014). *Understanding the masking-shadowing function in microfacet-based BRDFs.*
- Akenine-Moller, T., et al. (2018). *Real-time rendering 4th ed.*
- Pharr, M., et al. (2018). *Physically based rendering: From theory to implementation 4th ed.*

マイクロファセットモデル

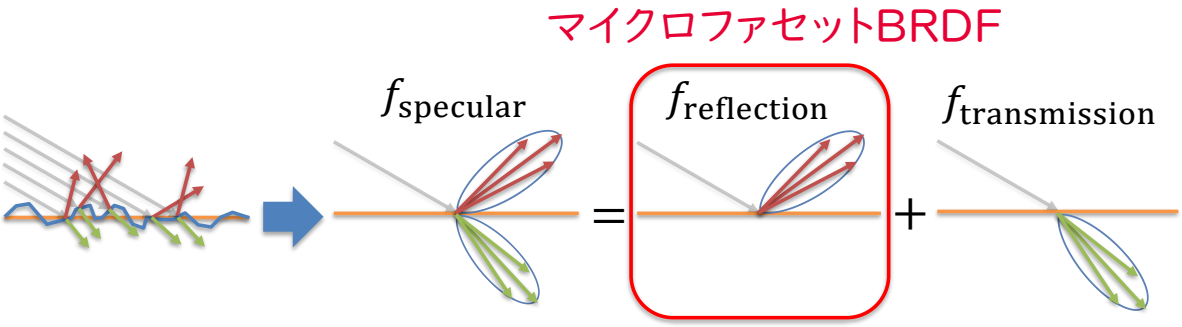
小さな表面の集まりで、粗い面を表現するモデル



まず、マイクロファセットモデルを紹介します。マイクロファセットモデルは、凸凹を持った小さな面(microfacet)を集めた microsurface と、凸凹を単純化した macrosurface です。

マイクロファセットモデル

小さな表面の集まりで、粗い面を表現するモデル



統計的な反射の分布とはどういう意味なのか、もう少し詳しく説明します。まず、凸凹の表面に光が入射して…

(click)(click)..

反射や屈折をすることを考えます。そして、

(click)

それら反射や屈折した光線の散らばり具合を、それぞれ分布関数として表現します。

(click)

さらに、その分布関数を反射成分と屈折成分に分けます。

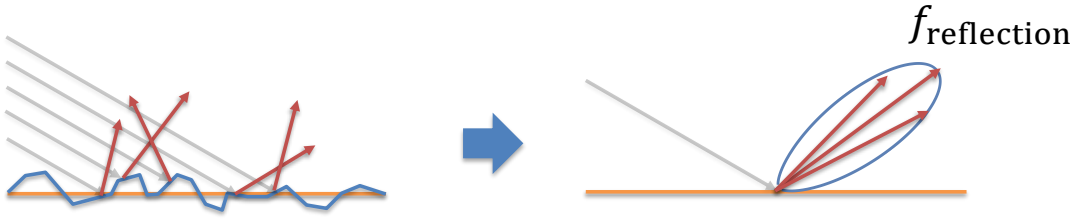
(click)

このとき、反射成分をマイクロファセットBRDFと呼びます。ここでは、マイクロファセットBRDFのみに注目して説明します。

マイクロファセット BRDF

小さな**鏡面**の集まりで、粗い面を表現するモデル

$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$



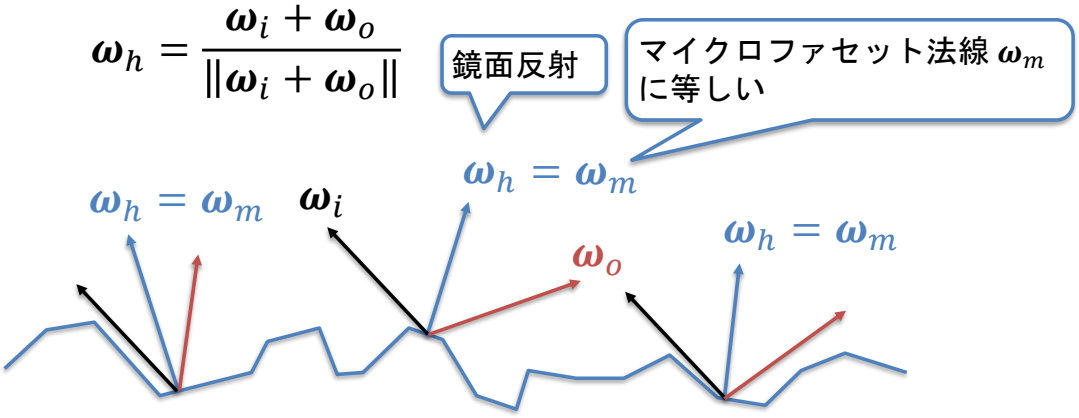
マイクロファセットBRDFとは、小さな鏡面の集まりで、粗い面を表現するモデルです。大まかに、「F」「G_2」「D」という関数が組み合わさって成り立っています。ここでは、それぞれの関数の意味を説明します。

ハーフベクトル

$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$

ハーフベクトル $\boldsymbol{\omega}_h$

$$\boldsymbol{\omega}_h = \frac{\boldsymbol{\omega}_i + \boldsymbol{\omega}_o}{\|\boldsymbol{\omega}_i + \boldsymbol{\omega}_o\|}$$



はじめに、ハーフベクトルという量を導入します。ハーフベクトルとは…

(click)

入射方向と

(click)

反射方向の

(click)

真ん中の方向を表すベクトルです。これは、

(click)

鏡面反射が起きるときの法線になっています。

このベクトルが

(click)

マイクロファセットの法線に等しいとします。

(click)

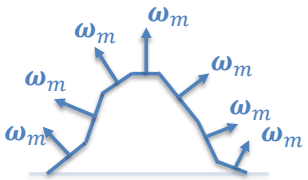
これにより、マイクロファセットは法線 ω_m をもつ鏡面として考えることになります。

法線分布関数

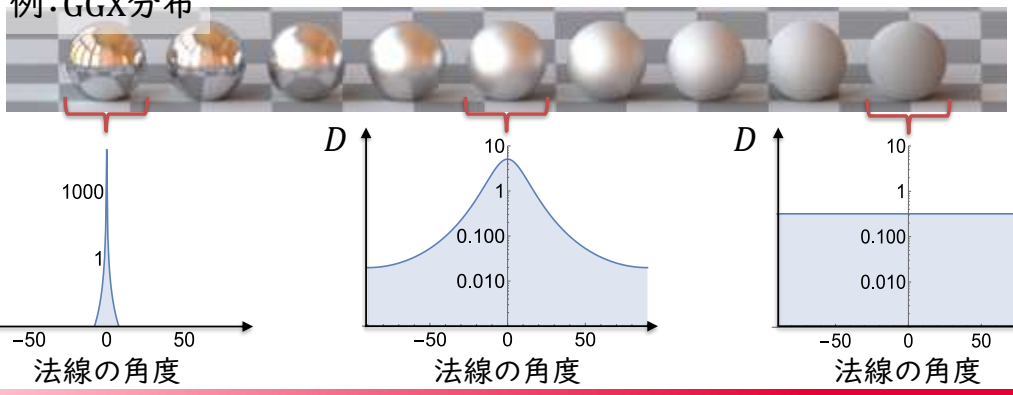
$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$

法線分布関数 $D(\boldsymbol{\omega}_m)$

– 光反射に関わる面積分布



例：GGX分布



CEDEC2024

次に、法線 ω_m に基づく、法線分布関数を紹介します。法線分布関数は、光反射に関わる面積の分布を表します。

(click)

GGX分布を例にとると、こんな感じです。

(click)

ツルツルな面では、法線のほとんどは真上を向いています。

(click)

すこしざらざらになると、ほかの方向を向く面が増えます。

(click)

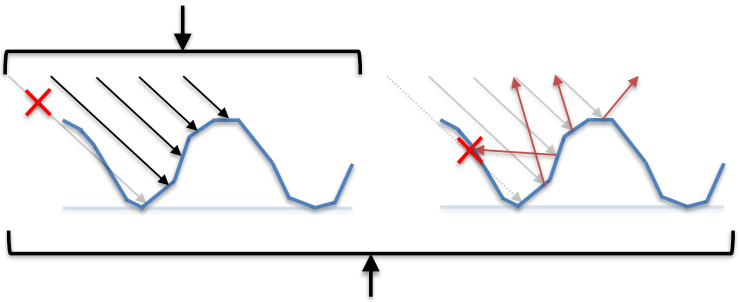
とてもざらざらなところでは、面は満遍なくいろいろな方向を向きます。

Masking関数

$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$

Smith Masking 関数 $G_1(\boldsymbol{\omega}, \boldsymbol{\omega}_m)$

– 他の表面からの遮蔽



Masking-Shadowing 関数 $G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m)$

– 入射方向 $\boldsymbol{\omega}_i$と出射方向 $\boldsymbol{\omega}_o$ に対する遮蔽

そして、Masking関数を紹介します。Masking関数 G_1 とは、ほかの表面からの遮蔽を表す量です。

(click)

一方向からの遮蔽のほかに、入射方向と反射方向の両方の遮蔽を表す、Masking-Shadowing 関数 G_2 があります。

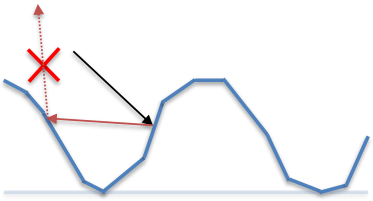
(click)

図は、Masking 関数のあり・なしを比較したものです。「Masking関数あり」のほうは、何の変哲もない、土でできたポットですが…
「Masking関数なし」のほうは、ポットのふちが不自然に光っています。
これは、表面で反射した光が遮蔽されなくなるぶん、過剰に光線が反射されることになるからです。

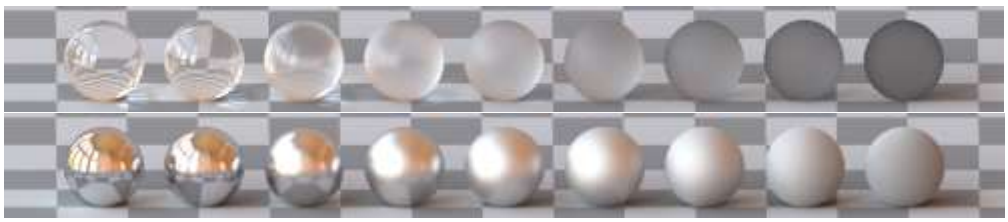
注意事項

$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$

$f_{\text{reflection}}$ は表面の1回反射を表現



ラフになるほど
反射光が不足



ここで注意することをお伝えします。それは、マイクロファセットBRDFは、表面上の一回反射を表すということです。前のスライドで見たように…

(click)(click)

表面で起きた反射は遮蔽されます。これはつまり

(click)

2回目、3回目の反射は考えないことを意味します。

これにより、

(click)(click)

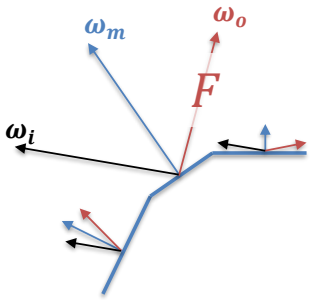
ラフになるほど、反射光が足りなくなることになります。表面上の多重反射を考慮しない限り、この点はマイクロファセットBRDFの制約事項として注意する必要があります。

Fresnel項

$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$

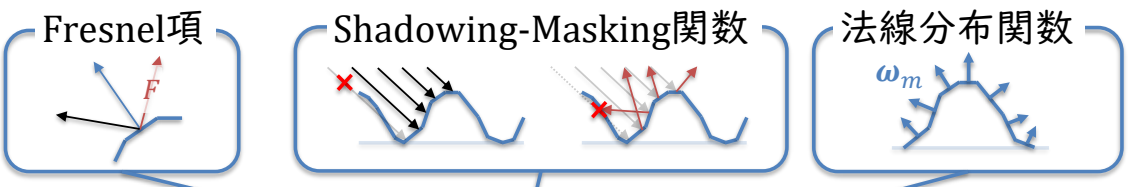
Fresnel項 $F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m)$

– 物質ごとの表面反射特性の違い



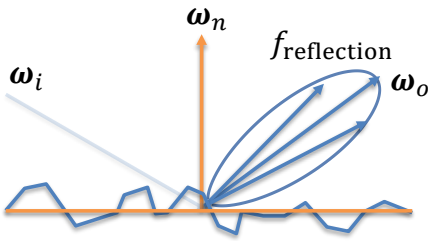
最後に、フレネル項 F を紹介します。これは、物体の境界で起こる光の反射について、屈折率や複素屈折率で表される材質ごとに異なる、表面反射特性の違いを表す量です。フレネルありの図では、絶縁体で起こる光反射について、ポットの反射を水平方向から眺めたとき、反射率が高くなる現象がおきることがわかります。フレネルなしの図では、反射率が一定、ここでは4%の設定した一定の反射率で、ポットの反射が不自然に暗くなることが見て取れます。

マイクロファセットBRDF まとめ



$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$

※積分の変数変換の係数など

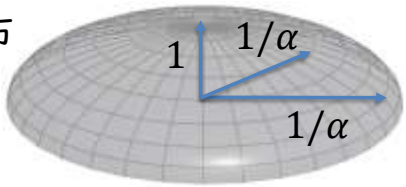


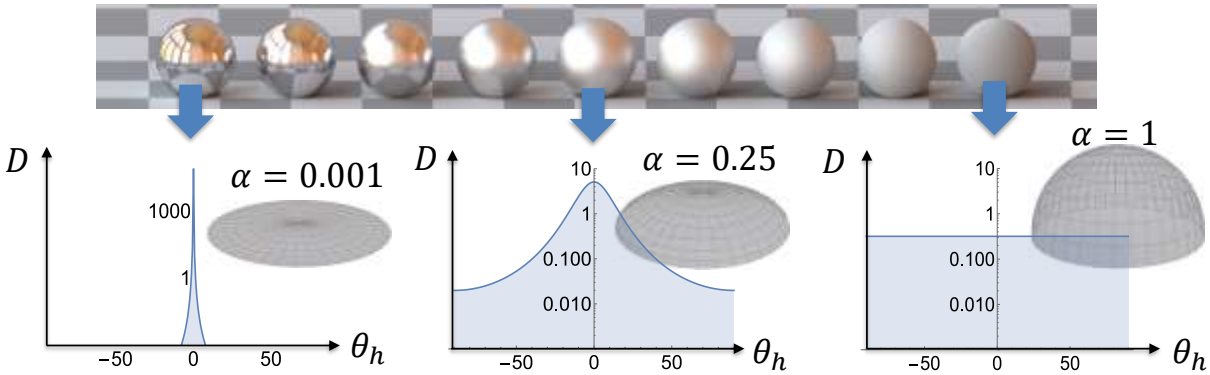
マイクロファセットBRDFのまとめです。マイクロファセットBRDFは、凸凹な表面上で起こる1回反射を表します。F は物質ごとの表面反射特性を担うFresnel項、G_2 は遮蔽を担うShadowing-Masking関数、D は面積分布を担う法線分布関数です。分母の $4|\omega_o \cdot \omega_n||\omega_i \cdot \omega_n|$ は積分の変数変換などによって得られる係数ですが、この講演では説明を割愛しています。

等方性GGX法線分布関数

- 赤道半径 $1/\alpha$ の回転楕円体の面積分布

$$D(\boldsymbol{\omega}_m) = \frac{1}{\pi} \frac{\alpha^2}{\cos^4 \theta_h (\alpha^2 + \tan^2 \theta_h)^2}$$





CEDEC2024

後半のパートに入る前に、具体的な法線分布関数を一つ紹介します。それは、物理ベースレンダリングの分野で最も有名なもののひとつである、GGX法線分布関数です。

GGXは、赤道半径 $1/\alpha$ の回転楕円体の面積分布に対応する法線分布関数です。

(click)(click)

ツルツルのところ (click) $\alpha = 0.001$ のつぶれた回転楕円体、

(click)(click)

少しラフになる $\alpha = 0.25$ では、少し立ち上がった回転楕円体、

(click)(click)

とてもラフになる $\alpha = 1$ では、半球の回転楕円体の面積分布に対応します。GGXを使った物理ベースマテリアルに現れる「Roughness」は、一般的に α を換算したものが使われています。このため、Roughness の値を変えることは、この回転楕円体の面積分布を変えることに対応しています。

参考文献

- Trowbridge, T. S. and Reitz, K. P. (1975). *Average*

irregularity representation of a rough surface for ray reflection.
- Walter, B., et al. (2007). *Microfacet models for refraction through rough surfaces.*
- Walter, B., et al. (2016). *The ellipsoid normal distribution function. Supplementary material.*
- Jonathan, D. and Anis, B. (2023). *Sampling Visible GGX Normals with Spherical Caps.*

後半パート

- はじめに
- 基礎知識
  - マイクロファセット (Microfacet)
  - 法線分布関数 (Normal Distribution Function)
  - マスキング関数 (Masking-Shadowing Function)
  - 可視法線分布関数 (Visible Normal Distribution Function)
- 最近の話題
  - Microsurface Transformationsの解説
    ヤコビアンさえあれば良い (Jacobian Is All You Need)
- 最後に

私のパートでお話しする内容はこのようになっていて、まず、下準備として基本的なことについて軽く説明し、それから最近の研究であるMicrosurface Transformationsを私の解釈を交えてできるだけわかりやすく解説したいと思います。元の論文はなかなか複雑ですが、特にヤコビアンだけわかっていればGGXを実装する際に必要となるサンプリングや確率密度関数の計算が簡単になる、ということを伝えたいと思っています。

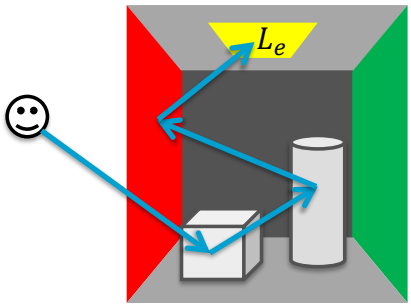
後半パート

$$L(\boldsymbol{\omega}_o) = L_e + \int_{\Omega^{\pm}} f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)\, L_{in}(\boldsymbol{\omega}_i) |\boldsymbol{\omega}_n \cdot \boldsymbol{\omega}_i|\, d\boldsymbol{\omega}_i$$

レンダリング方程式をパストレーシングで解く

GGXを使う時に必要となるのは

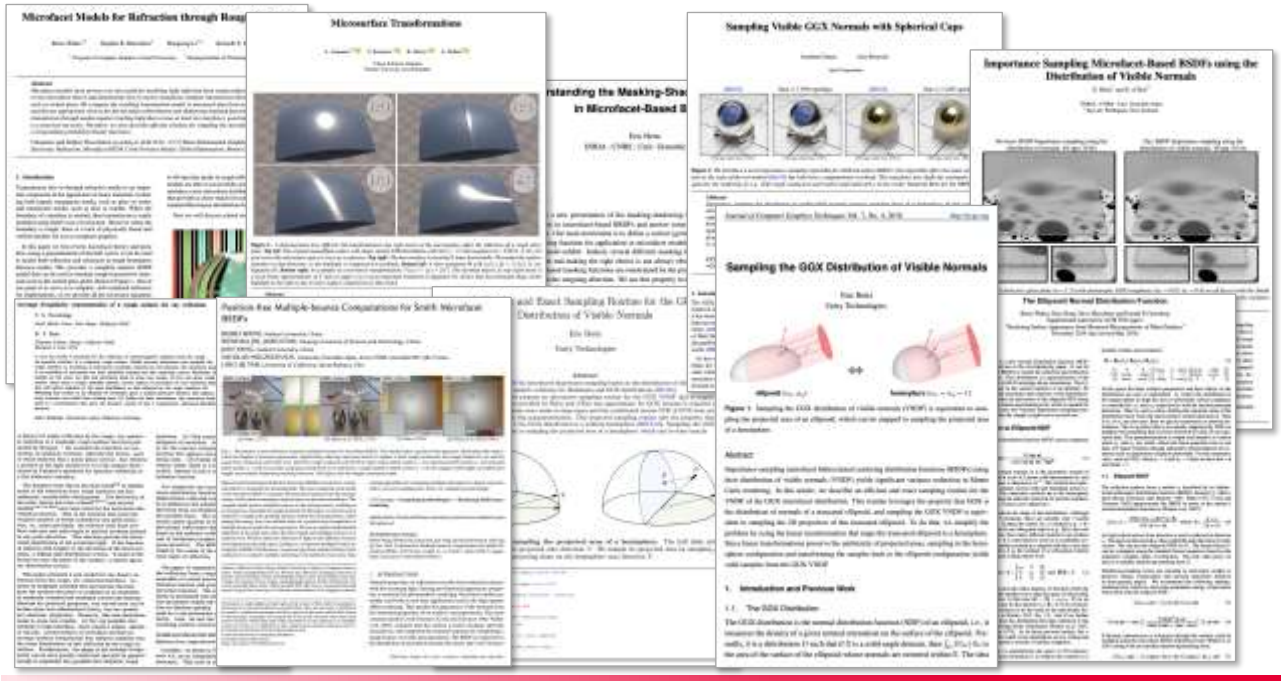
- サンプリング
- 確率密度関数



前半のパートはBRDFの成り立ちや評価についてでしたが、後半では、レンダリングの支配的な手法となりつつあるパストレーシングでGGXを用いるという状況を想定していて、サンプリングや確率密度関数の計算が主なテーマとなります。これらをどうやって簡単に行うかをお伝えするのが目的となっています。

はじめに

では早速始めましょう。



GGXの論文、たくさん出ていてややこしいですよね？ここに載せているのは一部でまだまだあります。金属やプラスチックなどの光沢をちょっとレンダリングしたいだけなのに、なんでこんなにたくさんの文献を読まなければいけないのか、と思ったことはないでしょうか？私も全く同じです。

はじめに

一つの単純な疑問から

今日はどうすれば少しでも楽に実装できるのかを理解してもらうため、
一つの極めて単純な質問から始めたいと思います。

はじめに

以下の関数はどんな形か？

$$D(\boldsymbol{\omega_m}) = \frac{1}{\pi\alpha_x\alpha_y\left(\frac{\omega_{m_x}^2}{\alpha_x^2}+\frac{\omega_{m_y}^2}{\alpha_y^2}+\omega_{m_z}^2\right)^2}$$

(α_xとα_yは縦方向と横方向での粗さ)

その質問というのは、ここに書かれたGGXの法線分布関数と呼ばれるものですが、これを見て一体どのような形か想像がつくでしょうか？というものです。私は全くイメージできませんでした。

はじめに

試しに$\alpha_x = \alpha_y = 1$を代入してみる

$\|\boldsymbol{\omega_m}\| = 1$なので

$$D(\boldsymbol{\omega_m}) = \frac{1}{\pi\left(\omega_{m_x}^2 + \omega_{m_y}^2 + \omega_{m_z}^2\right)^2} = \frac{1}{\pi}$$

こういった場合私はどうするかというと、具体的な数字を代入してみます。２つのパラメータαx、αyがあるので、試しに両方に１を入れてみましょう。するとなんと$\frac{1}{\pi}$という形になります。これは偶然ではありません。

はじめに

この単純化された$D(\boldsymbol{\omega_m})$を

$$D_{std}(\boldsymbol{\omega_m}) = \frac{1}{\pi}$$

と書くことにする

この単純化された法線分布関数をD_{std}と書くことにします。

はじめに

$D_{std}(\boldsymbol{\omega_m})$を使うと全てが簡単になる!

後ほど詳しくみていきますが、実はD_{std}だけ知っておくとサンプリングやPDFの計算、その理解が非常に楽になります。

はじめに

- 法線分布関数が質感を決める重要な役割を果たす
- GGXを使いこなすにはJacobianさえ分かれば良い

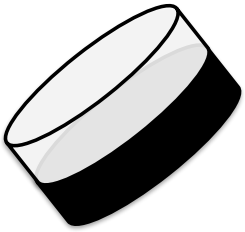
数式が出てきてすぐに分からなくても全く問題ないですが、この２点だけ記憶に留めておいていただくと、マイクロファセットやそれにまつわる新しい研究を理解する助けになるのではないかと思います。

Microfacet
マイクロファセット

最近の話題を紹介するにあたって予備知識が必要となるので、いくつか見ていきましょう。まずはマイクロファセットからです。

マイクロファセット

一つ一つの微小な面は
「鏡」
とよく書かれている



上側のみ鏡、下側は完全な黒
(Mono-sided microflakes)

マイクロファセット理論では、物体の表面を細かいマイクロファセットと呼ばれる面で表現し、その向きを統計的な分布で表すことで、さまざまな質感を再現することを容易にします。一般的な文献ではマイクロファセットは、完全鏡面であるとしています。今回話す内容では、マイクロファセットはdouble-sided materialで、上側のみ完全鏡面反射面で、下側、つまり、マクロな面を向いている側は完全な黒であると仮定します。これはmono-sided microflakesとも呼ばれます。

マイクロファセット

Heightfieldのみ扱える

珊瑚のような形状は扱えない

また古典的なマイクロファセットモデルはハイトフィールドは扱えますが、珊瑚のような入り組んだ形状は扱えません。

マイクロファセット

粗さを１としたときのGGXで表されるマイクロサーフェスの形状は、マイクロファセットを並べ替えると、

マイクロファセット

このように綺麗な半球へと変換できます。

Normal Distribution Function
法線分布関数

CEDEC2024

続いて法線分布関数についてみていきましょう。マイクロファセットの向きの分布を表すのが法線分布関数と呼ばれる関数です。英語ではNormal Distribution Function、略称NDFです。

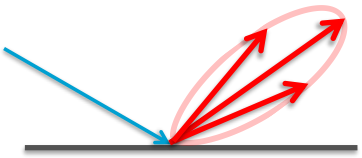
法線分布関数

$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$

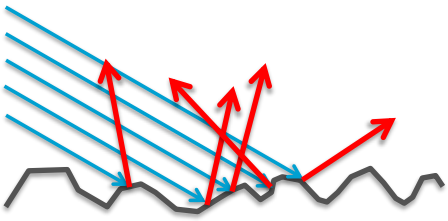
前半部分で出てきたBRDFの式のこの赤で強調された部分で通常Dを用います。

法線分布関数

$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$



$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$は
反射した光線の相対的な
強さの分布を表す



反射する光線の向きは
法線分布関数$D(\boldsymbol{\omega}_m)$によって決まる

BRDFは反射する光線の相対的な強さの分布を表しますが、微小面の向きを決める法線分布関数は、その分布形状の大部分を決定づける大切な役割を果たします。

法線分布関数

$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$

また、マスキング関数は法線分布関数によって決まりますし、

法線分布関数

$$f_{\text{reflection}}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_m) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m)}{4|\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_n||\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_n|}$$

フレネルもマイクロファセットの面の向きを用いて計算するので、ここにも影響を及ぼします。

法線分布関数

⚠ 球面上の分布と思いがち

気をつけて欲しいのは、この法線分布関数は球面上で定義された確率密度関数と誤解されがちなんですが、実はそうではないということです。

法線分布関数

⚠ 球面Ω上で積分しても1にならない

$$\int_{\Omega^{\pm}} D(\boldsymbol{\omega_m}) d\boldsymbol{\omega_m} \neq 1$$

残念ながら、これは通常正規化されておらず、球上で積分しても1になりません。

法線分布関数

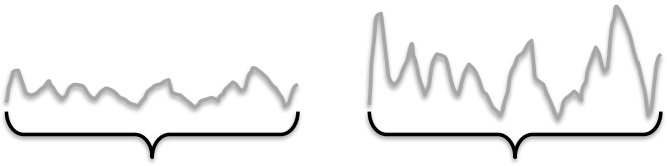
$\boldsymbol{\omega_m}$の方向を向いている
マイクロファセットの面積

(単位は$\mathbf{[m^2/sr]}$)

なぜかというと、Dは方向$\boldsymbol{\omega_m}$を向いている面の面積を表すものだから
です。

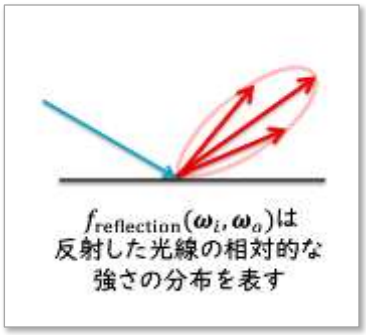
法線分布関数

凹凸が激しい場合には大きな値

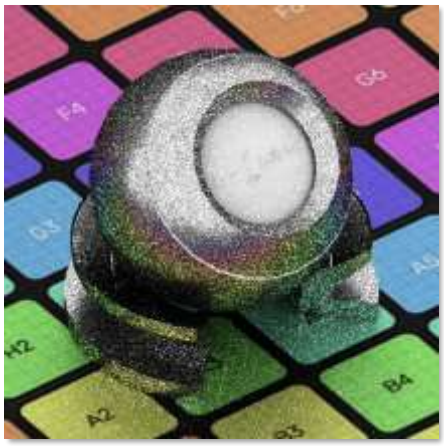


従って、似たような形状であったとしても、凹凸が激しいほど値が大きくなり、このままでは扱いづらいです。

法線分布関数



※ 光線の方向を決める
ことを**サンプリング**という



全方向に光線を投げた
256rays/pix

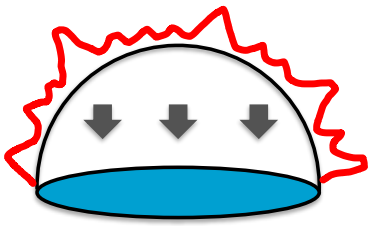
法線分布関数に従い光線を投げた
256rays/pix

先ほどのBRDFの図を少し思い返してみましょう。BRDFの値が小さくなる方向に光線を飛ばしても、反射の計算にあまり役立ちません。出鱈目な方向に光線を投げて反射の計算をした場合には効率が悪く真ん中のようにノイズだらけの画像となってしまい、法線分布関数に従って光線を投げた場合には、右のように綺麗な反射の像を得ることができます。このように、法線分布関数に従って光線を飛ばすには、法線の方向を決めるための、言い換えれば、サンプリングをするための確率密度関数が必要になります。

法線分布関数

確率密度関数にするにはペタンとつぶす

$$\int_{\Omega^{\pm}} D(\boldsymbol{\omega_m}) d\boldsymbol{\omega_m} \neq 1$$



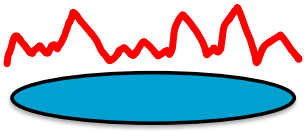
半球面で定義されたヒストグラム

実は、反球面上で定義された法線の向きのヒストグラムをマクロサーフェスの法線方向にペタンと潰すことで、どんなハイトフィールドの法線分布でも確率密度関数へと変換することができます。

法線分布関数

確率密度関数にするにはペタンとつぶす

$$\int_{\Omega^{\pm}} D(\boldsymbol{\omega_m})\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n}\rangle d\boldsymbol{\omega_m} = 1$$

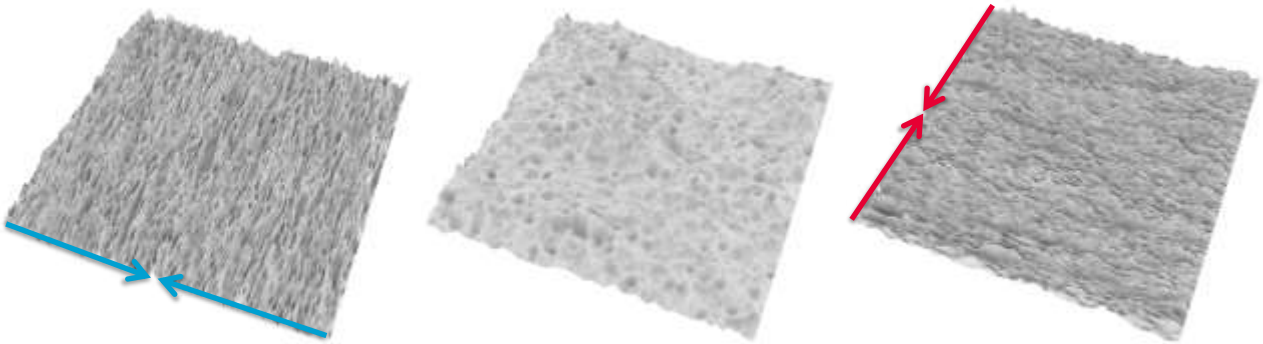


※ ヒストグラムの値にコサインをかけている

ハイトフィールドは潰してもそれを構成する面同士が重なることはありません。ですので、凹凸が激しい場合も、それほどではない場合も、潰してしまえば、ヒストグラムの総和はマクロサーフェスの単位面積と一致することになります。このつぶす操作というのは内積を一つかけるだけなので、$D(\boldsymbol{\omega_m})\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n}\rangle$が確率密度関数として機能するようになります。粗さが１の場合の標準法線分布関数の場合はヒストグラムが平らなので、文字通り半球をペタンと平面に潰すことになります。

法線分布関数

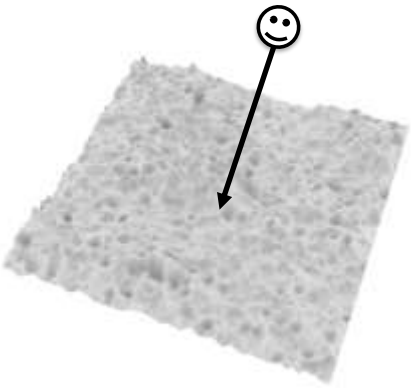
$$D(\boldsymbol{\omega_m}) = \frac{1}{\pi \alpha_x \alpha_y \left(\frac{\omega_{m_x}^2}{\alpha_x^2} + \frac{\omega_{m_y}^2}{\alpha_y^2} + \omega_{m_z}^2 \right)^2}$$



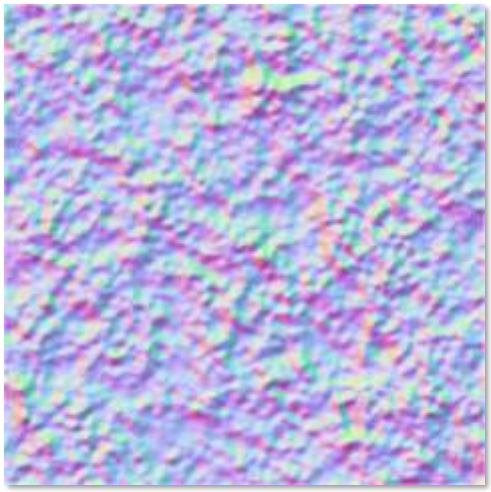
CGソフトでHeightfieldを生成

式ばっかりでは面白くありませんので、ハイトフィールドから確率密度関数をどう作るかみてみましょう。まずCGソフトでハイトフィールドを生成します。この凹凸は物体表面を顕微鏡で拡大したときの様子であると思ってください。この凹凸の程度が異なれば当然ながら違った材質に見えます。GGXの法線分布関数で現れるα_xとα_yはいわば縦横それぞれの潰れ度合いを表していて、潰れるほど大きな値となり、引き延ばされるほど小さくなります。潰す操作や引き延ばす操作というのは、線形変換でできます。

法線分布関数

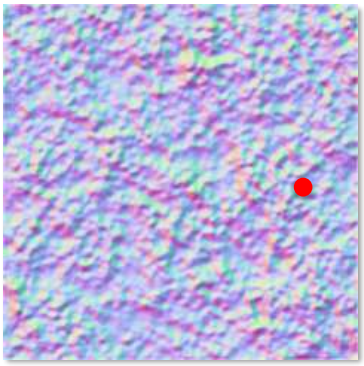


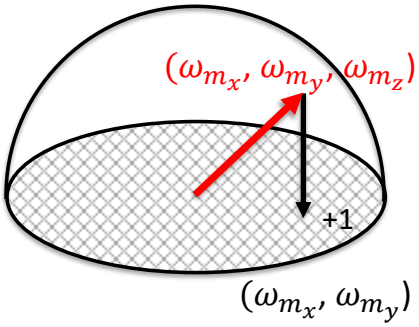
凹凸を真上から見てノーマルマップを生成



先ほど作った凹凸の面の向きに関する分布が欲しいので、ジオメトリを真上から見てノーマルマップを生成してみます。このノーマルマップからどの向きに向いている面が多いかというのを知るためのヒストグラムを作ります。実は真上から見るというのがマクロサーフェスの方向に沿って「潰す」という操作に相当します。

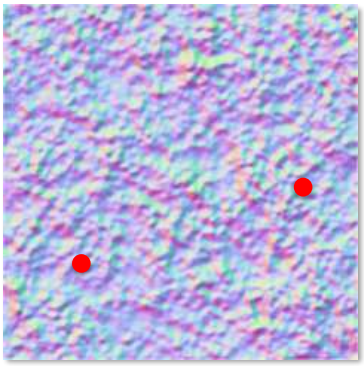
法線分布関数

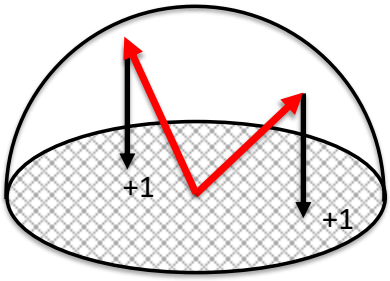




やり方はいろいろありますが、ここでは過去の論文と似たような図を作るために、ディスク形状のヒストグラム、濃淡画像を作ってみます。ノーマルマップ内の１つのピクセルを選び、その方向をディスクに投影して、その場所の場所のカウンタを１つ増やします。単位ベクトルを扱っているので、向きをエンコードするにはx座標とy座標だけがわかれば構いません。

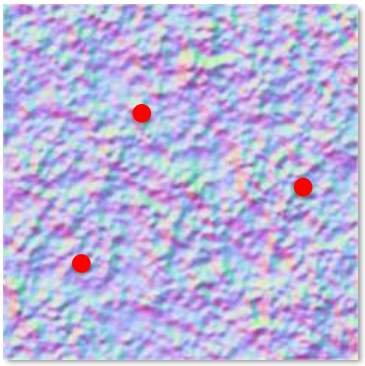
法線分布関数

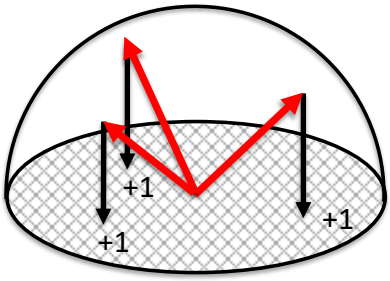




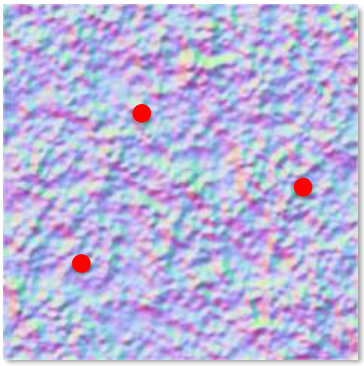
同様の操作をノーマルマップの全ての画素に対して行います。

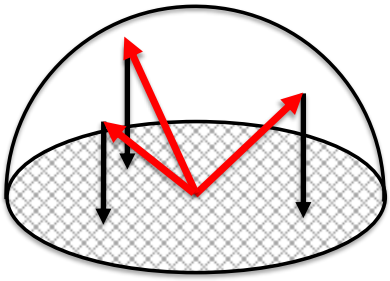
法線分布関数



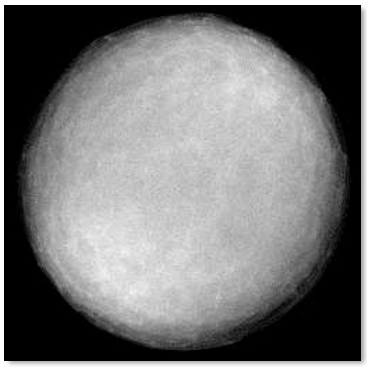


法線分布関数





$D(\boldsymbol{\omega_m})\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle$

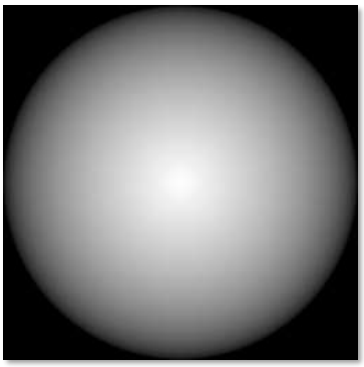


ヒストグラムの総和を画素数
で割ると1.0になっている

続けていくと、最終的に右の画像のようなヒストグラムが完成し、これを確率密度関数としてレンダリングに用いることができます。

法線分布関数

$D_{std}(\boldsymbol{\omega_m})\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle$



標準法線分布関数の場合

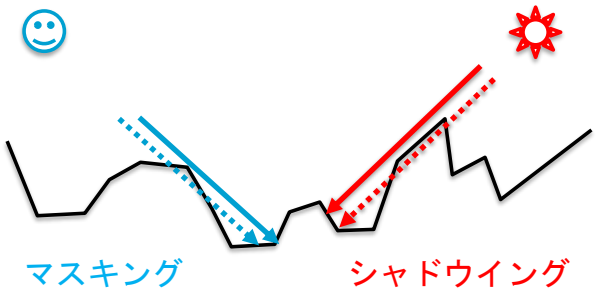
粗さ１のGGXの場合はこのような綺麗な分布となります。以上が簡単な法線分布関数の話でした。

Masking-Shadowing Function
マスキング関数

CEDEC2024

続いてマスキング関数について復習していきたいと思います。

マスキング関数



前半でもありましたが、マスキング関数(masking-shadowing function)というのは、どのくらい視線が遮れずに届くか、また、どのくらい光源からの光が遮られずに届くか、を表します。

マスキング関数

可視法線分布関数を
使う時に楽なので
分離可能の形式で考える

$$G_2 = G_1 G_1$$

MaskingとShadowingの２つを加味した関数をG_2としそれを使っても良いのですが、実際の計算においては、その２つをG_1 x G_1といった２つの積の形で表現した方が便利なことが多いです。なので、ここでもG_1の積の形で表せる、という仮定を採用します。

マスキング関数

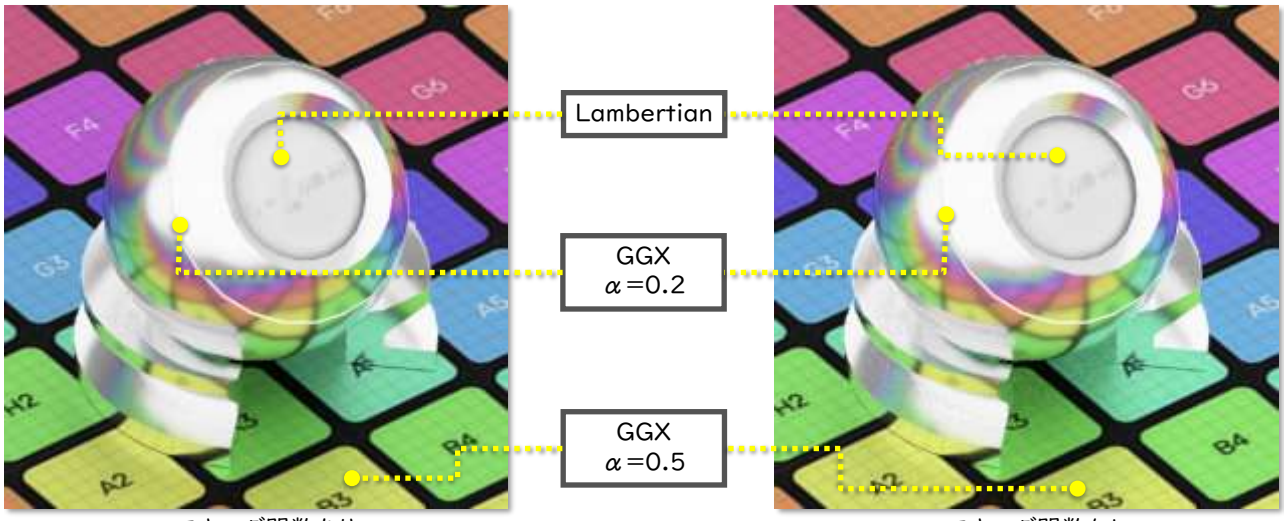
$$G_{1,\,std}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \chi^+(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o}) \frac{2\left|\omega_{o_z}\right|}{1 + \omega_{o_z}}$$

$\chi^+(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o})$

- $\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o}$が常に正となるため、BRDFの場合は不要
- 多くの場合クランプトコサイン$\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o} \rangle$と併用され不要

標準法線分布関数を用いている場合の標準マスキング関数はこの式で表されます。導出や最近の研究について配布資料で記述していますので、興味ある方は後ほど見てみてください。この式にはいくつか注意があって、まず$\chi^+(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o})$はBRDFの場合は不要です。それから、$\chi^+(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o})$は、クランプトコサイン$\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o} \rangle$と併用されるので多くの場合不要です。

マスキング関数

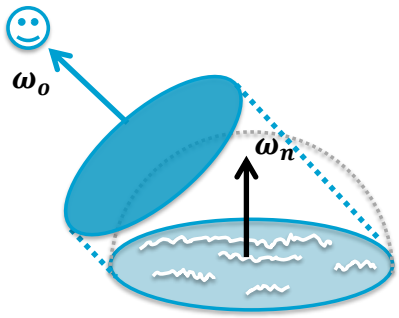


マスキング関数あり
256 samples/pixel

マスキング関数なし
256 samples/pixel

前半でもありましたが、マスキング関数がない場合は不自然に明るくなり、ノイズが発生してしまいます。

可視法線分布関数



$$|\cos\theta_o| = \int_{\Omega^{\pm}} G_{1,}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o})\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o}\rangle D(\boldsymbol{\omega_m}) d\boldsymbol{\omega_m}$$

さて、微小面を含む単位円を視点方向に投影することを考えます。視点から直接見えているマイクロファセットの面積の総和は$\boldsymbol{\omega_n}$と$\boldsymbol{\omega_o}$の内積$|\cos\theta_o|$で表すことができます。これは視点方向に投影されたマクロサーフェスの単位円の面積と同じです。凹凸があるので、投影した場合複数のマイクロファセットが重なってしまいますが、その重なりを取り除く役割をG_1が担います。

マスキング関数

$$|\cos\theta_o| = \int_{\Omega^{\pm}} G_1(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) \langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o} \rangle D(\boldsymbol{\omega_m}) d\boldsymbol{\omega_m}$$

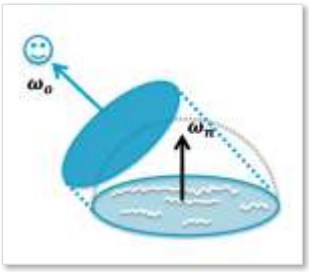
$$|\omega_{o_z}| = G_1(\boldsymbol{\omega_o}) \int_{\Omega^{\pm}} \chi^+(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o}) \langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o} \rangle D(\boldsymbol{\omega_m}) d\boldsymbol{\omega_m}$$

$$G_1(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \chi^+(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o}) \frac{|\omega_{o_z}|}{\int_{\Omega^{\pm}} \langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o} \rangle D(\boldsymbol{\omega_m}) d\boldsymbol{\omega_m}}$$

$$G_{1,\,std}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \chi^+(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o}) \frac{2|\omega_{o_z}|}{1 + \omega_{o_z}}$$

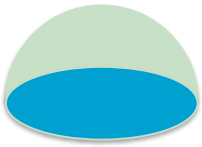
G_1 はマイクロファセットが視点方向に向いているのかをチェックするためのヘヴィサイド関数を括り出すと、残りの部分は視点方向$\boldsymbol{\omega_o}$にのみ依存するため、それを積分の外に出すことで、先ほどの形を導くことができます。導出はちょっと無理やりな気がしますが、この式が持つ幾何的な意味を考えてみましょう。

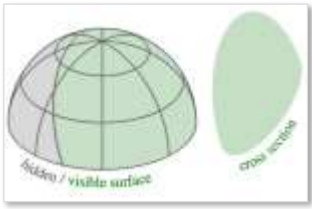
マスキング関数



ディスクの投影

$$\chi^+(\boldsymbol{\omega_m}\cdot\boldsymbol{\omega_o})\frac{\pi|\omega_{o_z}|}{\pi\int_{\Omega^\pm}\langle\boldsymbol{\omega_m},\boldsymbol{\omega_o}\rangle D_{std}(\boldsymbol{\omega_m})d\boldsymbol{\omega_m}}$$



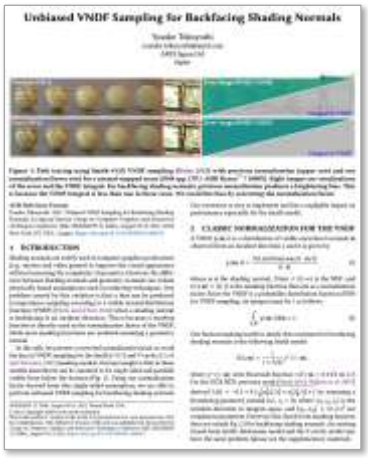


上半球の投影

CEDEC2024

右の図のように上半球の見える部分のみを視線方向に投影したものをcross sectionと呼びますが、先ほど出てきた式の分母と分子にπをかけると、マスキング関数は実はディスクの投影面積を、cross sectionの面積で割ったものとなっています。$\boldsymbol{\alpha}$を１とした標準の場合を考えることで、マスキング関数の幾何学的な意味が分かりやすくなりました。

マスキング関数



画像は“Unbiased VNDF Sampling for Backfacing Shading Normals”より

$$G_{1,std}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \chi^+(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o}) \frac{2|\omega_{o_z}|}{\omega_{o_z} + 1}$$

$$\frac{G_1(\boldsymbol{\omega_m}, \boldsymbol{\omega_o})}{H(\boldsymbol{\omega_o})} = \chi^+(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o}) \frac{2|\omega_{o_z}|}{\omega_{o,z} + \sqrt{\boldsymbol{\alpha_x}^2 \boldsymbol{\omega_{o_x}}^2 + \boldsymbol{\alpha_y}^2 \boldsymbol{\omega_{o_y}}^2 + \boldsymbol{\omega_{o_z}}^2}}$$

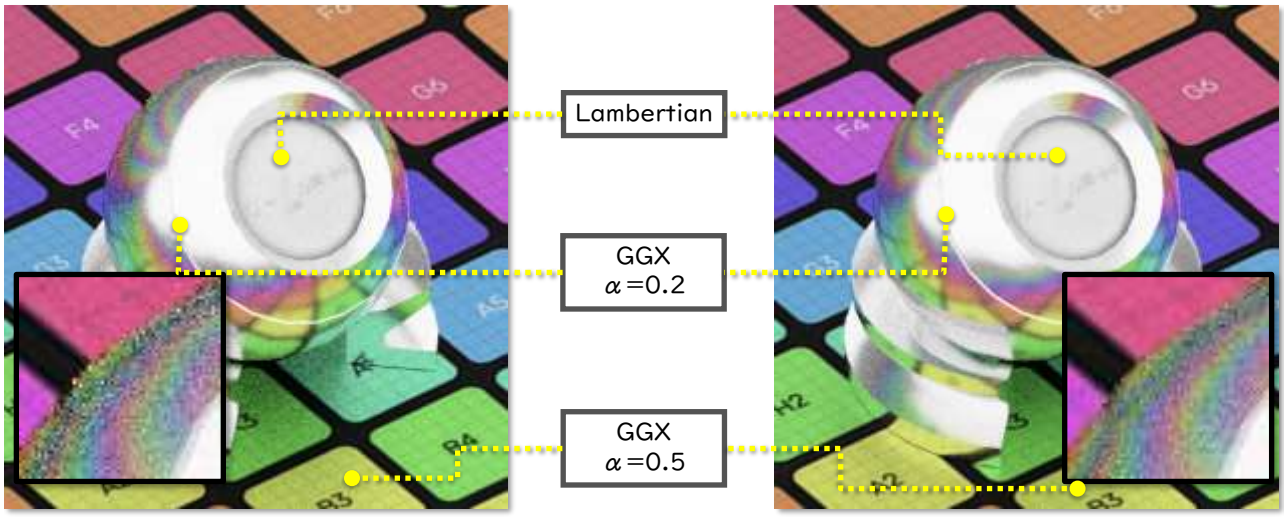
実は先ほど求めたマスキング関数は、AMDの徳吉さんがノーマルマップ適用時のために提案した正規化項と一致していて、何も特別なことをしなくても同じ結果が得られることを示しています。

Visible Normal Distribution Function
可視法線分布関数

CEDEC2024

続いて可視法線分布関数VNDFについて少しだけお話しします。これまでは視点の位置などを一切考慮しない法線分布関数の話でした。法線分布関数を用いてももちろんさまざまな凹凸を持つ面を表現できますが、実際サンプリングするときにマイクロファセットの面の向きを求めても、それが遮蔽されていたり、視点の反対を向いていたりすれば、せっかくの計算が無駄になります。なので、視点の方へ向いていて、かつ、遮蔽もされていないマイクロファセットのみからサンプリングを行いたいという要求が生じます。可視法線分布関数というのは(マスキングもされず)見えているマイクロファセットの向きの分布を表すため、それを用いることで無駄な計算をしなくて済むようになります。

可視法線分布関数

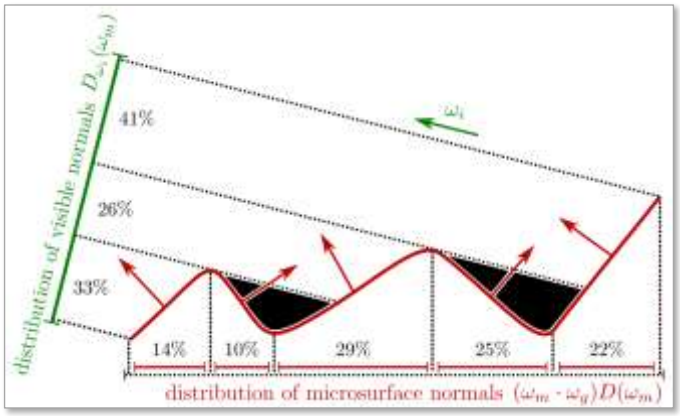


法線分布関数を使った
16 samples/pixel

可視法線分布関数を使った
16 samples/pixel

ここではどのような効果があるかのみお見せし、詳細は配布資料に含めるので興味のある方は後で読んでみてください。左は法線分布関数を用いた計算結果で、右は可視法線分布関数を用いて計算した結果です。私のPCではどちらも0.5sec程度ですが、可視法線分布関数で計算した方が綺麗な絵になっています。ただ、他のサンプリング方法と組み合わせて使用した場合はこのメリットが薄れてしまうこともあります。

可視法線分布関数

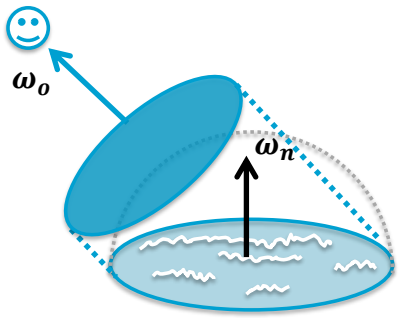


画像は“Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs”より

$$D_{vis}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \frac{G_1(\boldsymbol{\omega_m}, \boldsymbol{\omega_o})\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o}\rangle D(\boldsymbol{\omega_m})}{|\boldsymbol{\omega_n} \cdot \boldsymbol{\omega_o}|}$$

その可視法線分布関数ですが、どのように導出されるでしょうか？
この図で$\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o}\rangle$は見ている方向へ投影する役割、赤はマイクロファセット、横の破線はマクロサーフェス、緑の線は投影されたマクロサーフェスを表します。**分母は面積を表すので、絶対値を忘れないようにしましょう。**この式は、単位面積あたりの投影されたマイクロファセットの総面積は、重なりを除外すると、投影されたマクロサーフェスの面積と等しいということを意味しています。マスキング関数G_1は重なりによって多重に面積が加算されるのを防ぎます。

可視法線分布関数

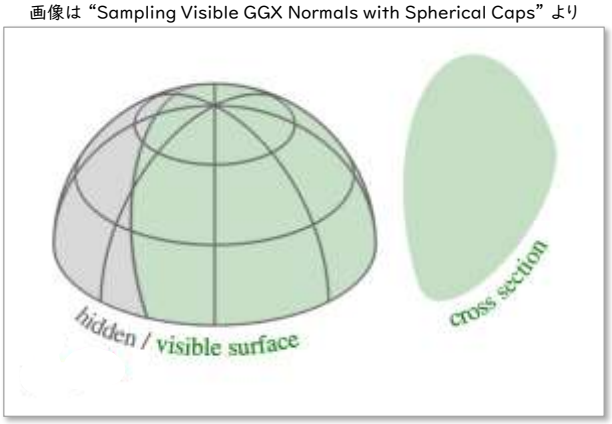


$$D_{vis,std}(\omega_m, \omega_o) = \frac{G_1(\omega_m, \omega_o)\langle \omega_m, \omega_o \rangle D_{std}(\omega_m)}{|\omega_n \cdot \omega_o|} = \frac{G_1(\omega_m, \omega_o)\langle \omega_m, \omega_o \rangle}{A_{projected\ disc}}$$

先ほど出てきた図と式を少し整理してみます。微細な凹凸を持つ半径1の円盤を見ていると考えましょう。このとき、重なりを除いて視点方向に投影されたマイクロファセットの総面積は単位円の面積πにω_nとω_oの内積をかけたものになります。青の斜めの楕円の面積ですね。各マイクロファセットを投影した面積にマスキング関数をかけたものを、このprojected discの面積で割ってやることで、確率密度関数が定義できます。重なりを除いてるので、これは見えているマイクロファセットの法線の分布を記述する関数となり、可視法線分布関数と呼ばれます。

可視法線分布関数

画像は“Sampling Visible GGX Normals with Spherical Caps”より



ところで、切断された楕円の視点から見えている面のみを視点方向に投影した面積、つまり、この図の右側の部分をcross sectionと呼びますが、可視法線分布関数からのサンプリングはこのcross sectionを一様にサンプリングすることで行うことができることが知られています。しかし、なぜそうなるのかは自明ではないため導出してみましょう。

可視法線分布関数

$$D_{vis,\,std}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \frac{G_{1,\,std}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o})\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o}\rangle D_{std}(\boldsymbol{\omega_m})}{|\boldsymbol{\omega_n} \cdot \boldsymbol{\omega_o}|}$$

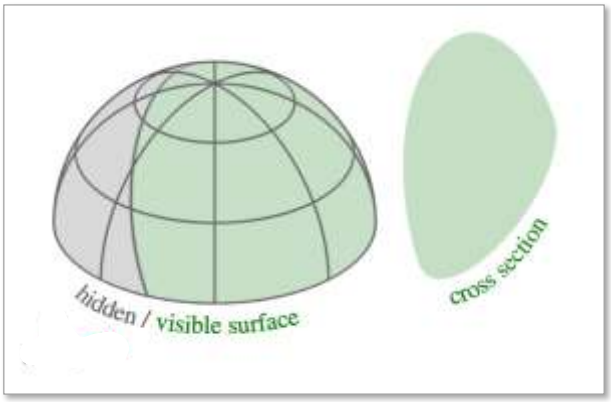
$$G_{1,\,std}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \frac{|\omega_{o_z}|\chi^+(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o})}{\int_{\Omega^\pm}\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o}\rangle D_{std}(\boldsymbol{\omega_m}) d\boldsymbol{\omega_m}}$$

$$D_{vis,\,std}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \frac{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o}\rangle D_{std}(\boldsymbol{\omega_m})}{\int_{\Omega^\pm}\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o}\rangle D_{std}(\boldsymbol{\omega_m}) d\boldsymbol{\omega_m}} = \frac{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o}\rangle}{\boldsymbol{A_{cross\ section}}}$$

先ほど導出したマスキング関数を元の可視法線分布関数の定義式に代入すると、いくつかの項が相殺されてcross sectionの面積を使った単純な式が得られます。

可視法線分布関数

画像は“Sampling Visible GGX Normals with Spherical Caps”より



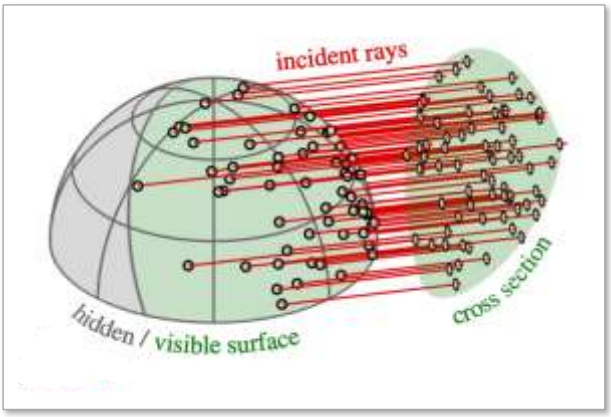
$$D_{vis,\,std}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \frac{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o} \rangle}{\boldsymbol{A_{cross\ section}}}$$

CEDEC2024

Lambertianを思い出してみてください。BRDFはCos/piという形で表されますが、サンプリング時には、単位円、面積piを均等にサンプリングし、その点を半球へと投影して方向を得るという手順が用いられます。

可視法線分布関数

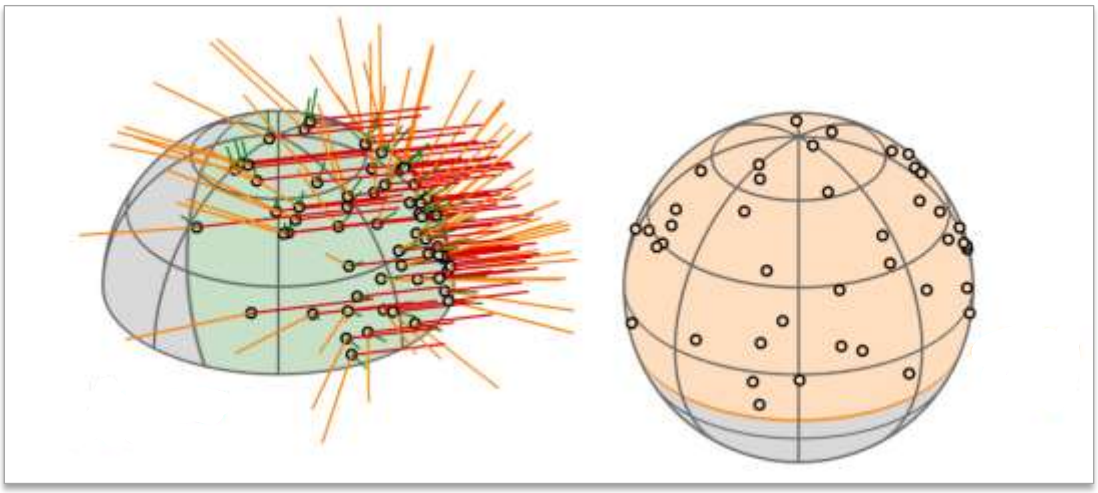
画像は“Sampling Visible GGX Normals with Spherical Caps”より



$$D_{vis,\,std}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \frac{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_o} \rangle}{\boldsymbol{A_{cross\ section}}}$$

ここでもcross sectionを均等にサンプリングして、半球へと投影するという同じ手順で方向を得ることができるわけです。

可視法線分布関数



画像は“Sampling Visible GGX Normals with Spherical Caps”より

最新の論文“Sampling Visible GGX Normals with Spherical Caps”では、もう一歩進んで、cross sectionを均等にサンプリングして得られた半球面上の点で反射したベクトルは、spherical cap上で均一に分布するという性質を利用し、さらに計算効率を上げています。

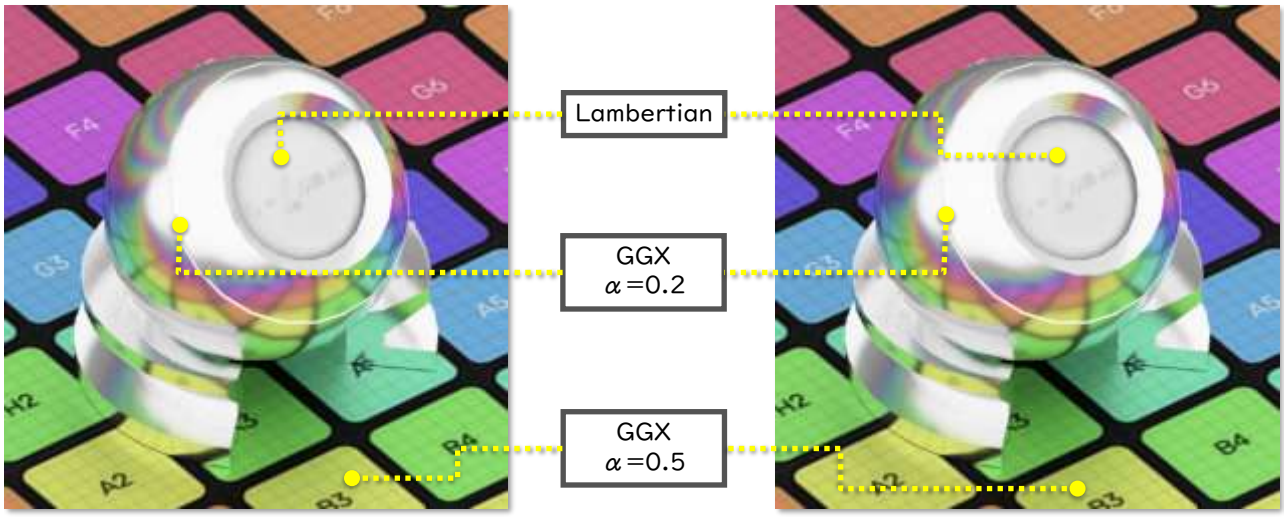
Microsurface Transformations

Jacobian Is All You Need
ヤコビアンさえあれば良い

CEDEC2024

さて、本題に入ります。ここからはMicrosurface Transformationsの内容をなるべく簡素化してお話ししたいと思います。少し込み入った話が続きますが、最終的に得られる計算は実はとっても簡単で、機械学習の有名な論文になぞらえれば、Jacobianさえ知っていれば良いですよ、といった内容になっています。

Microsurface Transformations



Walter (2007)
256 samples/pixel

Microsurface Transformations
256 samples/pixel

先にレンダリングした絵をお見せします。左はWalterの2007年の論文の方法で計算したもの、右はこれから紹介するMicrosurface Transformationsを使って計算したもので、全く同じ結果が得られます。ただし、ずっと理解でしやすい方法で計算することができます。

標準法線分布関数

$$\int_{\Omega^{\pm}} D(\boldsymbol{\omega_m})\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n}\rangle d\boldsymbol{\omega_m} = 1$$

確率密度関数にするには$\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n}\rangle$が必要だった

先ほどあった確率密度関数の話を思い出してみましょう。$D(\boldsymbol{\omega_m})$を確率密度関数にするには$\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n}\rangle$が必要でした。

標準法線分布関数

$$\int_{\Omega^{\pm}} D_{std}(\boldsymbol{\omega_m})\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n}\rangle d\boldsymbol{\omega_m} = \int_{\Omega^{\pm}} \frac{1}{\pi}\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n}\rangle d\boldsymbol{\omega_m} = 1$$

(これはLambertianを思い起こさせる)

余談になりますが、標準法線分布関数の確率密度関数は完全拡散反射面のLambertianのBRDFと酷似しています。LambertianのBRDFは反射ベクトルについての式ですが、ここに示した式はマイクロファセットの面の向きに関する式となっています。

標準の場合の計算手順

	マイクロファセット ω_m	反射/屈折したレイ ω_i
サンプリング	単位円から一点を選ぶ (concentric disc map)	• 点を半球面に投影 • 反射/屈折の計算
PDF	$\frac{1}{\pi}$	Jacobianを2つかける

標準の場合についてサンプリングとPDFの計算をまとめます。サンプリングは次のように行います。まず、単位円からランダムに１点を選び、それを半球面に投影して向きを求めます。それを用いて反射あるいは屈折したベクトルを求めて完了です。必要となるJacobianは、単位円から半球面に変換するのに必要なJacobianと、求めた向きで反射/屈折を計算するときに必要となるJacobianの２つです。

Jacobian ①投影

⚠ 標準法線分布関数 D_{std} で
生成した方向を $\boldsymbol{\omega_{std}}$ と呼ぶ

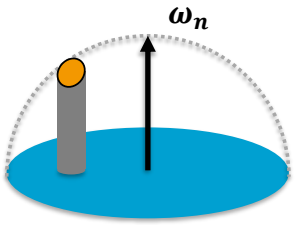
ここからは説明の上で明確に区別するため、標準法線分布関数 D_{std} で生成した向きを $\boldsymbol{\omega_{std}}$ と呼び、一般の場合のDで生成された向きに相当するものを引き続き$\boldsymbol{\omega_m}$と呼ぶことにします。

Jacobian ①投影

単位円上のサンプルから球面上のサンプルへ

$$\int_{Disc} D_{std}(\boldsymbol{\omega_{std}})\,dA = \int_{\Omega^{\pm}} D_{std}(\boldsymbol{\omega_{std}}) \left\| \frac{dA}{d\boldsymbol{\omega_{std}}} \right\| d\boldsymbol{\omega_{std}}$$

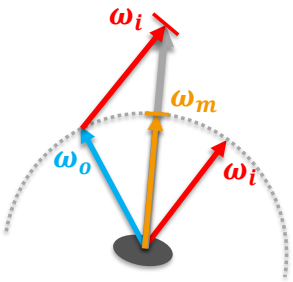
Jacobian



$$dA = d\boldsymbol{\omega_{std}} \langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle$$

先ほども話したようにω_{std}は、単位円を一様にサンプリングして、生成した点を半球面に投影することで、方向に変換することで得られます。非常に簡単ですね。今、半球面に投影するという処理が入ったので、変数変換を考慮するためのJacobianが必要になりますが、これは面積と立体角の関係から簡単に求まります。

Jacobian ②反射

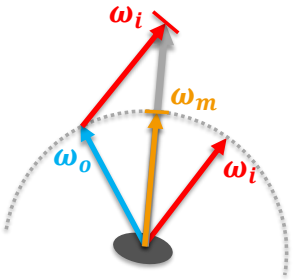


ω_mから反射/屈折したベクトルへ

$$\omega_i = 2(\omega_m \cdot \omega_o)\omega_m - \omega_o$$

ここで得られたω_mはマイクロファセットの向きですので、そのままレンダリングには使えません。入射したレイを、マイクロファセット面で反射あるいは屈折させてあげる必要があります。まずは反射させる場合から見ていきましょう。反射した向きはこのように単純な計算で求まります。

Jacobian ②反射



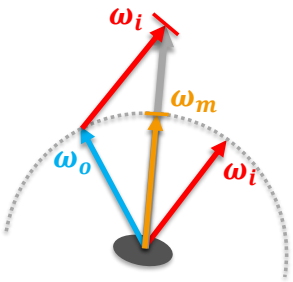
反射/屈折したベクトルのPDFが欲しい

$$\int_{\Omega^{\pm}} D(\boldsymbol{\omega_m})\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle \left\| \frac{\mathrm{d}\boldsymbol{\omega_m}}{\mathrm{d}\boldsymbol{\omega_i}} \right\| \mathrm{d}\boldsymbol{\omega_i} = 1$$

$\boldsymbol{\omega_m}$が動いた場合当然反射したベクトルも動きますが、動く量というのは同じではありません。それを加味するために確率密度関数の計算では、色がついている部分つまりJacobianが必要となります。

Jacobian ②反射

マイクロファセットで反射したベクトル



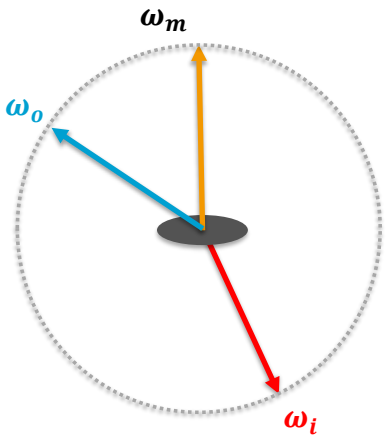
$$\omega_m = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|}$$

$$d\omega_m = \frac{|\omega_m \cdot \omega_o|}{\|\omega_i + \omega_o\|^2} d\omega_i = \frac{1}{4|\omega_m \cdot \omega_i|} d\omega_i$$

$$\left\| \frac{d\omega_m}{d\omega_i} \right\| = \frac{1}{4|\omega_m \cdot \omega_i|}$$

それは立体角の関係からこのように求めることができます。詳しい証明はWalterの2007年の論文にあるのでそちらを参考にしてください。

Jacobian ②屈折



$$\omega_i = \frac{\omega_m\left(\omega_m\cdot\omega_o-\sqrt{\eta^2-(1-(\omega_m\cdot\omega_o)^2)}\right)-\omega_o}{\left\|\omega_m\left(\omega_m\cdot\omega_o-\sqrt{\eta^2-(1-(\omega_m\cdot\omega_o)^2)}\right)-\omega_o\right\|}$$

$$\left\|\frac{d\omega_m}{d\omega_i}\right\| = \frac{\eta^2|\omega_m\cdot\omega_i|}{\left((\omega_m\cdot\omega_o)+\eta(\omega_m\cdot\omega_i)\right)^2}$$

(ηは2つの媒質の相対屈折率)

CEDEC2024

屈折についても証明は省略します。反射ないしは屈折する場合のJacobianのどちらかと、先ほどの半球面に投影する分のJacobianを$\frac{1}{\pi}$にかけることでPDFを求めることができるようになりました。

一般の場合の計算手順

D_{std}は等方性で$\alpha = 1$を仮定

D_{std}でサンプリングした$\boldsymbol{\omega_{std}}$
を任意の(α_x, α_y)で使いたい

(α_xとα_yは縦方向と横方向での粗さ)

ここまでは標準の場合、$\alpha = 1$の場合についての話だったので、一般の場合には今お話しした方法と同じ方法で求められないのであれば意味がないのでは?と思うかもしれません。しかし、実は一般の場合もほとんど同じ手順でマイクロファセットの面の向きを得ることができます。

一般の場合の計算手順

一般の場合はどうすれば良いか

$$(1,1) \neq (\alpha_x, \alpha_y)$$

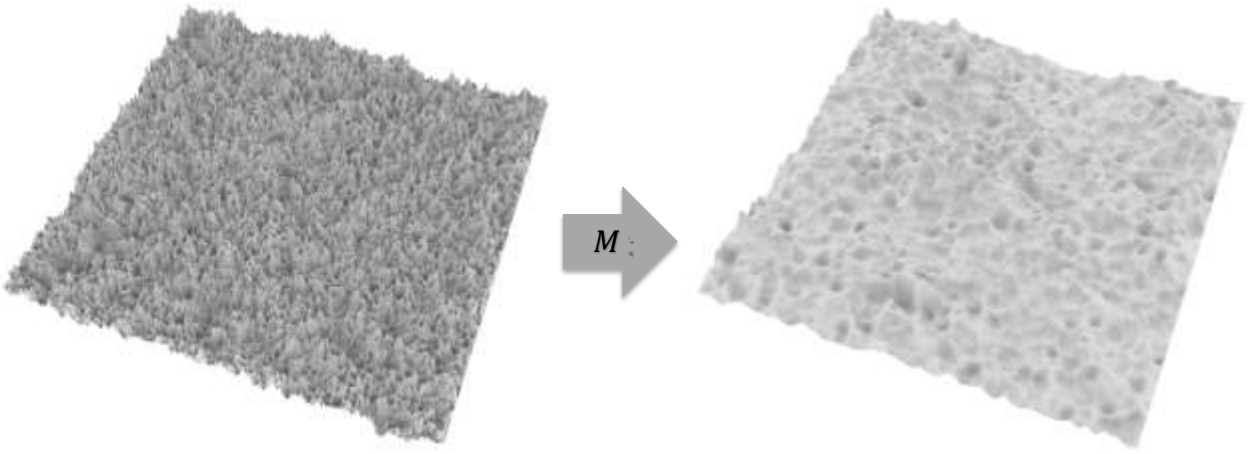
では、一般の場合α≠1の時はどうすれば良いでしょうか。サンプリングは向きに何か変換をかませば良いので簡単にできそうですが、確率密度関数の計算は少し難しそうですね。

一般の場合の計算手順

	マイクロファセット ω_m	反射/屈折したレイ ω_i
サンプリング	単位円から一点を選ぶ (concentric disc map)	• 点を半球面に投影 • 微小面の向きに線形変換 • 反射/屈折の計算
PDF	$\frac{1}{\pi}$	Jacobianを3つかける

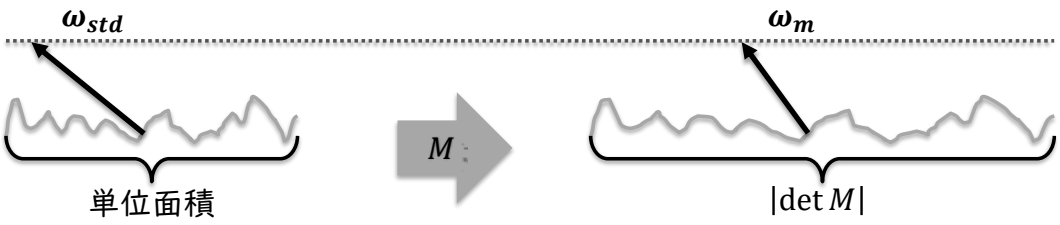
先ほどと同様に、一般の場合の手順を表にまとめます。サンプリングは標準の場合とほぼ同じで、まず単位円からランダムに１点を選び、それを半球面に投影して向きを求めます。次に任意のαに対応させるため、生成した方向ベクトルに線形変換を施します。それを用いて反射あるいは屈折したベクトルを生成して完了です。PDFの計算も標準の場合と大きく変わらず、生成したベクトルに施した線形変換に伴って、PDFを補正するため追加のJacobianが１つ必要となるだけです。

一般の場合の計算手順



まず引き延ばしについて見てみましょう。左のような凹凸を右のような少し滑らかなものに変換することを考えます。

一般の場合の計算手順



$$M = \begin{pmatrix} \frac{1}{\alpha_x} & * & \mathbf{0} \\ * & \frac{1}{\alpha_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

線形変換Mによって凹凸のある面を引き伸ばすと、マクロサーフェスの面積は$|\det M|$倍になり、引き伸ばされることで滑らかになるので光沢が増します。変換Mはマクロサーフェスに沿って行われるので、Z成分には作用せず、ベクトルのZ成分も当然ながら変化しません。

一般の場合の計算手順

マイクロファセット	変換	逆変換
ジオメトリ	M	M^{-1}
$\boldsymbol{\omega_m}$(法線) covector	M^{-T}	M^{T}

引き延ばされるのはマイクロサーフェスのジオメトリで、行列Mはジオメトリに対する変換を表します。$\boldsymbol{\omega_m}$はcovector、ここでは法線であるので、D_{std}から生成した$\boldsymbol{\omega_{std}}$にはMの逆転置行列をかけてあげる必要があります。

Jacobian ③引き伸ばし

画像は“Microsurface Transformations”より

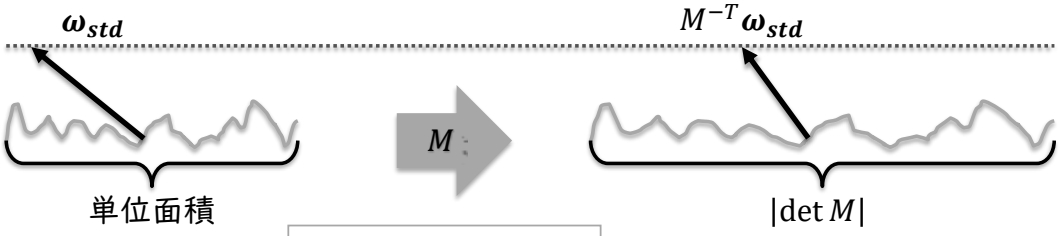
$\boldsymbol{\omega_{std}}$ と$\boldsymbol{\omega_m}$の関係は

$$D(\boldsymbol{\omega_m}) = \frac{|\det M|}{|M^T \boldsymbol{\omega_m}|^4} D_{std}(\boldsymbol{\omega_{std}})$$

サンプリングの追加処理はM^{-T} をかけるだけであり、いたって単純です。一方、PDFは少し複雑です。ここからは引き延ばしによって必要となるJacobianを詳しく見ていきましょう。まずD_{std}で生成したベクトル$\boldsymbol{\omega_{std}}$と$\boldsymbol{\omega_m}$の関係を知る必要があるわけですが、Microsurface Transformationsの論文によると、一般の場合のDと標準の場合のD_{std}の関係はこのように表されます。論文出は左の画像のように非常にややこしい式を経由してこの結果を得ています。分子の$|\det M|$については、マイクロサーフェスが引き伸ばされて面積が$|\det M|$に変わるので出所が想像できますが、分母については一見しただけではどういうものか分かりません。

Jacobian ③引き伸ばし

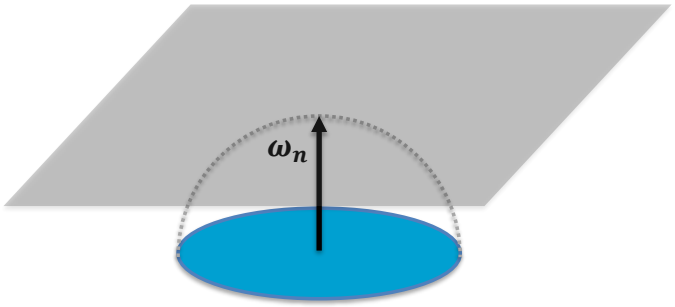
$$\int_{\Omega^{\pm}} D_{std}(\boldsymbol{\omega_{std}})\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n}\rangle \mathrm{d}\boldsymbol{\omega_{std}} = \int_{\Omega^{\pm}} D_{std}(\boldsymbol{\omega_{std}})\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n}\rangle \left\| \frac{\mathrm{d}\boldsymbol{\omega_{std}}}{\mathrm{d}\boldsymbol{\omega_m}} \right\| \mathrm{d}\boldsymbol{\omega_m}$$



$$\boldsymbol{\omega_m} = \frac{M^{-T}\boldsymbol{\omega_{std}}}{\|M^{-T}\boldsymbol{\omega_{std}}\|}$$

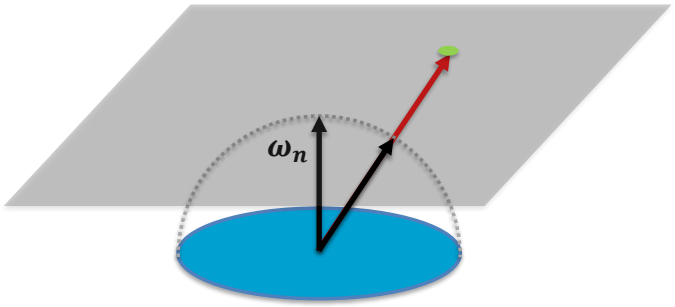
標準法線分布関数からサンプリングしたベクトルを線形変換しますが、ここでも変数変換を行なってみましょう。先ほど引き延ばしによるベクトルの変化はマクロサーフェスに沿って行われ、マクロサーフェスの法線方向の成分は変化しない述べました。ですので、引き延ばしによる立体角の変化を考えるには、スロープスペースを経由すると分かりやすくなります。$\boldsymbol{\omega_m}$は一番下にあるように引き伸ばしてから正規化します。

Jacobian ③引き伸ばし



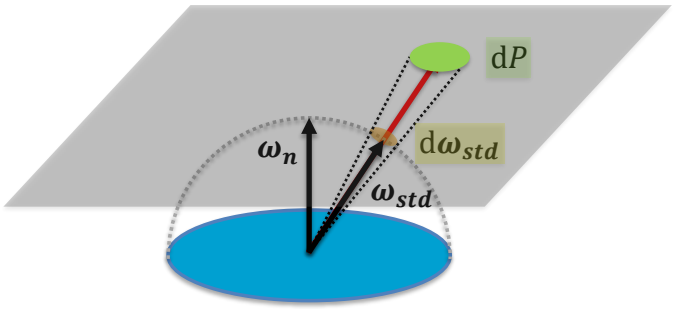
スロープスペースというと難しそうですが、そんなことはありません。半径１の半球に無限に続く平面が乗っている状況を思い浮かべてください。

Jacobian ③引き伸ばし



方向ベクトルを、この平面上の点の位置によって表現するのに使います。

Jacobian ③引き伸ばし

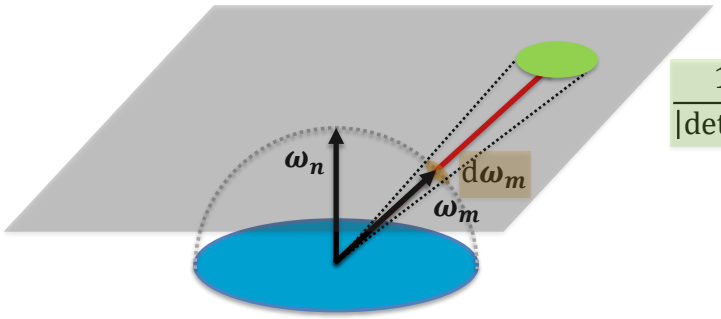


微小面積 dP と微小立体角 $d\omega_{std}$ の関係から

$$dP\langle\omega_{std},\omega_n\rangle\frac{1}{\frac{1}{\langle\omega_{std},\omega_n\rangle^2}}=dP\langle\omega_{std},\omega_n\rangle^3=d\omega_{std}$$

ここで平面上の微小な面積dPと微小な立体角$d\omega_{std}$の関係は内積の3条を使って表すことができます。dPが表す立体角を求めているのですが、面の傾きと距離の２条とを合わせると、内積の３条が出てきます。赤いベクトルはω_{std}が平面に当たるまで延長したもので、赤い項はそのベクトルの長さです。グレーの内積は微小面dPとω_{std}との向きにより求まります。

Jacobian ③引き伸ばし



$$\frac{1}{|\det M|}\mathrm{d}P$$

同様に微小立体角 $\mathrm{d}\boldsymbol{\omega_m}$ は引き延ばされた微小面積 $\frac{1}{|\det M|}\mathrm{d}P$ を用いて

$$\frac{1}{|\det M|}\mathrm{d}P\langle\boldsymbol{\omega_m}, \boldsymbol{\omega_n}\rangle^3 = \mathrm{d}\boldsymbol{\omega_m}$$

同様に微小な立体角 $\mathrm{d}\boldsymbol{\omega_m}$ は引き延ばされた微小な面積 $\frac{1}{|\det M|}\mathrm{d}P$ を用いて
このように表すことができます。

Jacobian ③引き伸ばし

$$\int_{\Omega^{\pm}} D_{std}(\boldsymbol{\omega_{std}})\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n}\rangle \left\| \frac{\mathrm{d}\boldsymbol{\omega_{std}}}{\mathrm{d}\boldsymbol{\omega_m}} \right\| \mathrm{d}\boldsymbol{\omega_m} = 1$$

微小面積 $\mathrm{d}P$ と微小立体角 $\mathrm{d}\boldsymbol{\omega_{std}}$の関係から

$$\mathrm{d}P\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n}\rangle^3 = \mathrm{d}\boldsymbol{\omega_{std}}$$

$\mathrm{d}P$ を引き延ばすので $\boldsymbol{\omega_m}$ に関しては

$$\frac{1}{|\det M|}\mathrm{d}P\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n}\rangle^3 = \mathrm{d}\boldsymbol{\omega_m}$$

代入してまとめていきましょう。ここにある変数変換は$D_{std}(\boldsymbol{\omega_{std}})$が$\frac{1}{\pi}$なので次のように書き換えることができます。

Jacobian ③引き伸ばし

$$1 = \int_{\Omega^{\pm}} \frac{1}{\pi} |\det M| \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^3}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^3} \langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle d\boldsymbol{\omega_m}$$

色付けしてみていきましょう。

Jacobian ③引き伸ばし

$$1 = \int_{\Omega^{\pm}} \frac{1}{\pi} |\det M| \frac{\langle \omega_{std}, \omega_n \rangle^3}{\langle \omega_m, \omega_n \rangle^3} \langle \omega_{std}, \omega_n \rangle d\omega_m$$

PDFは標準法線分布の$\frac{1}{\pi}$、引き伸ばしによる青色の項$\frac{\langle \omega_{std}, \omega_n \rangle^3}{\langle \omega_m, \omega_n \rangle^3}$の積、半球面に投影した内積$\langle \omega_{std}, \omega_n \rangle$、となっています。色付けされた３つの項の積は$\omega_m$のPDFなので、これに反射あるいは屈折の分のJacobianをかけるとPDFの導出は終了です。

Jacobian ③引き伸ばし

$$D(\boldsymbol{\omega_m}) = |\det M| \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^4}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^4} D_{std}(\boldsymbol{\omega_{std}})$$

実は先ほどの結果はMicrosurface Transformationsの論文の導出とは少し異なっています。論文ではこういった変換がなされています。

Jacobian ③引き伸ばし

$$\int_P D_{std}(\boldsymbol{\omega_{std}})\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n}\rangle \left\| \frac{\mathrm{d}\boldsymbol{\omega_{std}}}{\mathrm{d}P} \right\| \mathrm{d}P = 1$$

微小面積 $\mathrm{d}P$ と微小立体角 $\mathrm{d}\boldsymbol{\omega_{std}}$の関係から

$$\mathrm{d}P(\boldsymbol{\omega_{std}} \cdot \boldsymbol{\omega_n})^3 = \mathrm{d}\boldsymbol{\omega_{std}}$$

$$\int_P D_{std}(\boldsymbol{\omega_{std}})\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n}\rangle^4 \mathrm{d}P = \int_P P_{22,\,std}\mathrm{d}P = 1$$

$$P_{22,\,std} = D_{std}(\boldsymbol{\omega_{std}})\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n}\rangle^4$$

これは変数変換を$\boldsymbol{\omega_{std}}$から$\boldsymbol{\omega_m}$ではなく、スロープスペースである$\mathrm{d}P$へと行うことで求めることができます。先ほどと同様に微小面積 $\mathrm{d}P$ と微小立体角 $\mathrm{d}\boldsymbol{\omega_{std}}$の関係を用いると、スロープスペースの確立密度関数を導くことができます。

Jacobian ③引き伸ばし

$$\underbrace{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^4 \; D(\boldsymbol{\omega_m})}_{\text{スロープ空間 } P_{22}} = |\det M| \; \underbrace{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^4 \; D_{std}(\boldsymbol{\omega_{std}})}_{\text{スロープ空間 } P_{22,\,std}}$$

論文中の関係式

$$D(\boldsymbol{\omega_m}) = \frac{|\det M|}{|M^T \boldsymbol{\omega_m}|^4} D_{std}(\boldsymbol{\omega_{std}})$$

そのスロープスペースでの関係から、論文中の関係式に辿り着くことができます。

Jacobian ③引き伸ばし

$$\frac{1}{|M^T \boldsymbol{\omega_m}|^4} = \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^4}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^4}$$

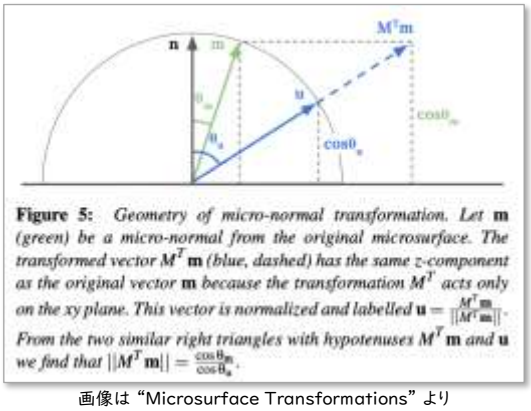


Figure 5: *Geometry of micro-normal transformation. Let* $\mathbf{m}$ *(green) be a micro-normal from the original microsurface. The transformed vector* $M^T\mathbf{m}$ *(blue, dashed) has the same z-component as the original vector* $\mathbf{m}$ *because the transformation* M^T *acts only on the xy plane. This vector is normalized and labelled* $\mathbf{u} = \frac{M^T\mathbf{m}}{||M^T\mathbf{m}||}$. *From the two similar right triangles with hypotenuses* $M^T\mathbf{m}$ *and* $\mathbf{u}$ *we find that* $||M^T\mathbf{m}|| = \frac{\cos\theta_\mathbf{m}}{\cos\theta_\mathbf{u}}$.

画像は“Microsurface Transformations”より

この二つは同じものなのですが、幾何学的な説明は論文にあるので、
興味のある方は見てみてください。

Jacobian ③引き伸ばし

$$D(\boldsymbol{\omega_m}) = |\det M| \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^4}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^4} D_{std}(\boldsymbol{\omega_{std}})$$

この関係式ですが、$D_{std}(\boldsymbol{\omega_{std}})$に$\frac{1}{\pi}$を代入して簡略化してみましょう。

Jacobian ③引き伸ばし

$$D(\boldsymbol{\omega_m}) = |\det M| \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^4}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^4} \frac{1}{\pi}$$

こうなります。

Jacobian ③引き伸ばし

$$D(\boldsymbol{\omega_m}) = |\det M| \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^4}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^4} \frac{1}{\pi}$$

を

$$1 = \int_{\Omega^{\pm}} D(\boldsymbol{\omega_m}) \langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle d\boldsymbol{\omega_m}$$

に代入

これを最初に出てきた積分の式に代入してみましょう。

Jacobian ③引き伸ばし

$$D(\boldsymbol{\omega_m}) = |\det M| \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^4}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^4} \frac{1}{\pi}$$

$$1 = \int_{\Omega^{\pm}} |\det M| \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^4}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^4} \frac{1}{\pi} \langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle d\boldsymbol{\omega_m}$$

これはさらに簡略化できます。

Jacobian ③引き伸ばし

$$D(\boldsymbol{\omega_m}) = |\det M| \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^4}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^4} \frac{1}{\pi}$$

$$1 = \int_{\Omega^{\pm}} \frac{1}{\pi} |\det M| \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^3}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^3} \langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle d\boldsymbol{\omega_m}$$

一部キャンセルされるので、先ほど求めた結果と一致します。めでたしめでたしとういうわけです。

マスキング関数

$$G_{1,\,std}(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = \chi^{+}(\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o}) \frac{2\left|\omega_{o_z}\right|}{1 + \omega_{o_z}}$$

ベクトルに適切な変換を施せば$G_{1,\,std}$がそのまま使える

$$G_1(\boldsymbol{\omega_m}, \boldsymbol{\omega_o}) = G_{1,\,std}\left(\frac{\boldsymbol{M^T\omega_m}}{\|\boldsymbol{M^T\omega_m}\|}, \frac{\boldsymbol{M^{-1}\omega_o}}{\|\boldsymbol{M^{-1}\omega_o}\|}\right)$$

- 線形変換変換を受けても遮られる部分は変化しない
  - 見えていた場所は見えたまま
  - 影になっていた部分は影のまま

マスキング関数については粗さが１の時の標準のマスキング関数をそのまま使うことができます。これはマイクロファセットに線形変換を施したとしても、見ている方向についても適切な変換を施してあげれば、全てが単純に引き伸ばされただけであるので、見えていた部分は見えているまま、影になっていた部分は影のままであるためです。

一般の場合の計算手順

- サンプリング
  - 単位円内の一点を(Concentric mapを用いて)選ぶ
  - 点を半球面に投影して方向に変えて $\boldsymbol{\omega_{std}}$ を得る
  - $\boldsymbol{\omega_{std}}$ に M^{-T} をかけて $\boldsymbol{\omega_m}$ を得る

- PDF
  - $\frac{1}{\pi}$ に3つのJacobianをかける
    - 半球面に投影した分: $\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle$
    - M^{-T} をかけた分: $|\det M| \frac{\langle \boldsymbol{\omega_{std}}, \boldsymbol{\omega_n} \rangle^3}{\langle \boldsymbol{\omega_m}, \boldsymbol{\omega_n} \rangle^3}$
    - 反射/屈折の分: $\frac{1}{4|\boldsymbol{\omega_m} \cdot \boldsymbol{\omega_o}|}$ など

サンプリングの手法と、PDFの計算をまとめてみます。とても簡単ですね。

一般の場合の計算手順

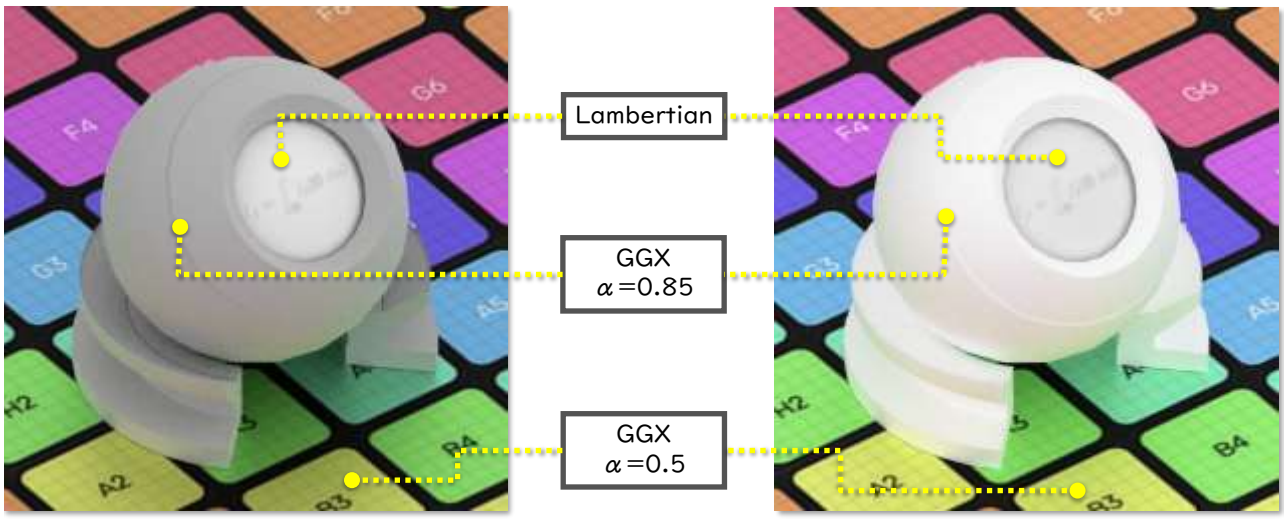
$$M = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$M = \begin{pmatrix} 10 & \mathbf{20} & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$M = \begin{pmatrix} 30 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

計算は理解しやすくなるものの、実際に高速化につながるわけではありません。ただ、理解しやすい以外にも実用上のメリットはあります。線形変換には対角成分以外に値があってもよく、メッシュの変形に追従して光沢を変化させることが容易になります。

一般の場合の計算手順



Microsurface Transformations
256 samples/pixel

Microsurface Transformations 多重散乱
256 samples/pixel

今回は紹介しませんでしたが、多重散乱も粗さ1の標準法線分布関数や標準マスキング関数を使って行えるので実装が簡単になります。粗さが大きい場合でも、右のように暗くなることなく自然な結果がえられます。

最後に

最後に

- マイクロファセット、NDF、VNDF、マスキング関数について復習した
- Microsurface Transformationsを用いると、異方性の場合や、Skewがかかった場合でも、GGXのNDFのサンプリングが容易になり、PDFも $\frac{1}{\pi}$ にJacobianを数個かけるだけで簡単に求められることを示した
- 論文では複雑な式が用いられていることが多いが、変数変換を施しわかりやすい空間に持ってくることで、簡略化できる

私のパートではGGXを中心にサンプリングとPDFの計算についてまとめ、論文のように複雑な計算をしなくとも同じ結果に辿り着けることをいくつかの例で示しました。論文を読んでそのまま理解するのは第一歩で、咀嚼するとその先に必ず大切な何かが見つかります。それは、本質的な疑問であったり、簡略化であったり、新しい手法であったりします。検索をするとBRDFについての解説記事は山ほど見つかりますが、できれば査読を経た論文をしっかり読んで、上辺だけではない知識をしっかりと身につけて行ってください。

参考文献

光学現象全般
[1999] Born, M. and Wolf, E. *Principles of Optics, 7th ed.*
[2014] 田所 利康, 石川 謙, 大津 元一. *イラストレイテッド光の科学.*
[2013] Zangwill, A. *Modern electrodynamics.*
[2015] Heckt, E. *Optics. 5th ed.*

物理ベースレンダリング
[2018] Akenine-Moller, T., et al. *Real-time rendering 4th ed.*
[2023] Pharr, M., et al. *Physically based rendering: From theory to implementation 4th ed.*
[2023] Adobe Inc. *THE PBR GUIDE - PART 1.*
[2023] Adobe Inc. *THE PBR GUIDE - PART 2.*

屈折率、複素屈折率、フレネル反射率
[2015] Babar, S. and Weaver, J. H. *Optical constants of Cu, Ag, and Au revisited.*
[2020] Hoffman, N. *Some Thoughts on the Fresnel Term.*

参考文献

マイクロファセット

[1975] Trowbridge, T. S. and Reitz, K. P. *Average irregularity representation of a rough surface for ray reflection.*
[2007] Walter, B., et al. *Microfacet models for refraction through rough surfaces.*
[2014] Heitz, E. *Understanding the masking-shadowing function in microfacet-based BRDFs.*
[2014] Heitz, E. and d'Eon, E. *Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals.*
[2016] Walter, B., et al. *The ellipsoid normal distribution function. Supplementary material.*
[2016] Heitz, E., et al. *Multiple-scattering microfacet BSDFs with the Smith model.*
[2016] Dupuy, J., et al. *Additional Progress Towards the Unification of Microfacet and Microflake Theories.*
[2017] Heitz, E. *A Simpler and Exact Sampling Routine for the GGX Distribution of Visible Normals.*
[2017] Ribardière, M., et al. *STD: Student's t-Distribution of Slopes for Microfacet Based BSDFs.*
[2017] Schüssler, V., et al. *Microfacet-based Normal Mapping for Robust Monte Carlo Path Tracing.*
[2018] Heitz, E. *Sampling the GGX distribution of visible normals.*
[2022] Atanasov, A., et al. *Microsurface Transformations.*
[2022] Tokuyosi, Y. *Unbiased VNDF Sampling for Backfacing Shading Normals.*
[2023] Jonathan, D. and Anis, B. *Sampling Visible GGX Normals with Spherical Caps.*
[2022] Bitterli, B. and d'Eon, E. *A Position-Free Path Integral for Homogeneous Slabs and Multiple Scattering on Smith Microfacets.*
[2022] Wang, B., et al. *Position-free Multiple-bounce Computations for Smith Microfacet BSDFs.*
[2023] Yuang Cui. *Multiple-bounce Smith Microfacet BRDFs using the Invariance Principle.*
[2023] d'Eon, E., et al. *Microfacet theory for non-uniform heightfields.*
[2023] d'Eon, E. *Student-T and Beyond: Practical Tools for Multiple-Scattering BSDFs with General NDFs.*

リソース

CGの生成に使ったレンダラー
前半パート
[2022] Jakob, W., et al. *Mitsuba 3 renderer. Version 3.1.1*
後半パート
[2023] 著者の開発したものによる

CGの生成に使ったモデルや環境マップ
[2016] Bitterli, B. *Rendering resources*
[2017] McGuire, M. *Computer Graphics Archive*

謝辞 (五十音順)

- 伊藤佑樹
- 内村 創
- 小澤祐樹
- 佐藤利音
- 嶋田知隼
- 鈴木 健太郎
- 高野修一
- 竹歳正史
- 平井健太朗
- 安田廉
- 山口崇

- 株式会社ポリフォニー・デジタルの皆さま